Verbs and Me: An Investigation Into Verbs as Labels for Action Recognition in Video Understanding

Michael Wray

A dissertation submitted to the University of Bristol in accordance with the requirements for the degree of Doctor of Philosophy in the Faculty of Engineering

December, 2019

58884 words

Abstract

This thesis focuses on the task of fine-grained action understanding in videos, specifically on the tasks of action recognition and action retrieval, with the aim of bridging the gap between language and vision. Typically, action segments were labelled with a (small) chosen set of verbs and/or nouns which are semantically unambiguous. This approach, called a closed vocabulary, doesn't allow for interesting relationships between the verbs to be discovered or utilised, as well as being unnatural when compared to that of a human's. This thesis explores the issues with expanding the vocabulary of verbs used for action understanding, including using an unbounded set.

For the action recognition task, videos are commonly given ground truth in the form of a verb and a noun. Semantic knowledge from external sources have successfully related nouns when the vocabulary size is increased from a closed vocabulary, but has been largely under-explored for verbs. This thesis aims to delve into this area in three ways: Firstly, open vocabulary annotations are collected from multiple annotators and related through the use of WordNet's verb hierarchy. Secondly, multi-verb, verb-only annotations are evaluated for the tasks of action recognition and action retrieval. Finally, this thesis presents the fine-grained action retrieval task which aims to relate videos and captions when they are semantically similar.

Declaration

I declare that the work in this dissertation was carried out in accordance with the Regulations of the University of Bristol. The work is original, except where indicated by special reference in the text, and no part of the dissertation has been submitted for any other academic award.

Any views expressed in the dissertation are those of the author and in no way represent those of the University of Bristol.

The dissertation has not been presented to any other University for examination either in the United Kingdom or overseas.

SIGNED: DATE:

Acknowledgements

Firstly, I would like to give thanks to my supervisor, Dima Damen, for her help and continued support throughout my PhD.

Thank you to all everyone in the Visual Information Laboratory: Austin, Jay, Teesid, Hazel, Ramon, Davide, Eduardo, Abel, Vangelis, Laurie, Ed, Will, Yanan, Jonny, Miguel, Xingrui, Young, Sam, Erik, Richard, Farnoosh, Perla, Fae, (blue) Sam, and Toby, you all made the lab a pleasure to work in. Your help and assistance made this thesis possible.

Thank you to everyone in Naver Labs Europe who was kind enough to discuss with me whilst I was there, but especially Diane and Gabriela who made sure my internship was interesting and fruitful both during and after I had left.

Finally, I would like to thank my mum, grandma and sister for their encouragement, and my girlfriend Sam for her support and understanding during the four years.

Publications

The work described in this thesis has been presented in the following publications:

1. Michael Wray, Davide Moltisanti, Walterio Mayol-Cuevas and Dima Damen. SEMBED: Semantic Embedding of Egocentric Action Videos. *European Conference on Computer Vision Workshops*, 2016.

2. Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price and Michael Wray. Scaling Egocentric Vision: The EPIC-KITCHENS Dataset. *European Conference on Computer Vision*, 2018.

3. Michael Wray and Dima Damen. Learning Visual Actions Using Multiple Verb-Only Labels. *British Machine Vision Conference*, 2019.

4. Michael Wray, Diane Larlus, Gabriela Csurka and Dima Damen. Fine-Grained Action Retrieval through Multiple Parts-of-Speech Embeddings. *International Conference on Computer Vision*, 2019.

For My Father.

Contents

List of Figures

List of Tables

Acronyms

ADL	Activities in Daily Living Dataset.
AH	Action Hyponyms, level of semantic relevancy.
AM	Action Meanings, level of semantic relevancy.
AMT	Amazon Mechanical Turk
AS	Action Synonyms, level of semantic relevancy.
AV	Action Verbs, level of semantic relevancy.
BEOID	Bristol Egocentric Object Interactions Dataset
BOW	Bag of Words
CMU	Carnegie Mellon University Multi-Modal Activity Dataset
CNN	Convolutional Neural Network
FC	Fully Connected (CNN layer)
FLOPs	FLoating point OPerations per second
FV	Fisher Vectors
GloVe	Global Vectors (Vectorised Word Representation)
GMM	Gaussian Mixture Model
GRU	Gated Recurrent Unit
GZS	Generalised Zero Shot
GTEA+	Georgia Tech Egocentric Activities Gaze + (Dataset)
HMDB	Human Motion Database
HMM	Hidden Markov Model
HoF	Histogram of Oriented Flow
HoG	Histogram of Gradients
IDT	Improved Dense Trajectories
IFV	Improved Fisher Vectors
IoU	Intersection over Union

KLT	Lucas and Kanade Tracker
K-NN	K-Nearest Neighbour
LCS	Least Common Subsumer
LSMDC	Large Scale Movie Description Challenge
LSTM	Long-Short Term Memory
MAC	Maximum Activations of Convolutions
mAP	mean Averaged Precision
MBH	Motion Boundary Histograms
MLP	Multi Layer Perceptron
MS-COCO	Microsoft Common Objects in COntext Dataset
MSR-VTT	Microsoft Research Visual-to-Text Dataset
MSVD	Microsoft Visual Description Dataset
MV	Multi-Verb (Labelling method)
MW	Markov Walk
PCA	Principle Component Analysis
RGB	Red Green Blue (colour model)
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SAMV	Soft-Assigned Mult-Verb (Labelling method)
SIFT	Scale Invariant Feature Transform
SfM	Structure from Motion
SLAM	Simultaneous Localisation and Mapping
SV	Single Verb (Labelling method)
SVG	Semantic Visual Graph
SVG$_u$	Undirected Semantic Visual Graph
SVO	Subject, Verb, Object
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency
TSN	Temporal Segment Networks
TT	Text-to-Text (for retrieval)
TV	Text-to-Video (for retrieval)
VN	Verb Noun (Labelling method)

VT	Video-to-text (for retrieval)
VV	Video-to-Video (for retrieval)
WuP	Path based similarity measure (Wu Palmer)
ZS	Zero Shot
ZSA	Zero Shot Actions
ZSN	Zero Shot Verbs
ZSV	Zero Shot Nouns

Chapter 1

Introduction

Language and vision represent two very different modalities of information. Yet, humans are able to transfer between them with little effort. Video understanding within action recognition is focussed almost entirely on non-overlapping class labels through the use of classification [9, 33, 133]. This is aided by the annotations collected for these datasets which come from a closed vocabulary [22, 60, 65, 124, 130]. Even when open vocabulary labels are collected, these are often converted to closed vocabulary labels via clustering [23] or majority voting [91]. Understanding the complex, overlapping space of open vocabulary actions, and specifically verbs, has not been attempted.

Information retrieval represents a form of video understanding more focussed on language. Instead of classes, videos are labelled with unique captions which use open natural language descriptions [144]. Powerful visual-language embeddings are learnt allowing for both within-modal and cross-modal retrieval to be performed. However, the importance of data for this task cannot be underestimated. In order to successfully perform cross-modal retrieval a large number of examples are required with the most recent datasets including over 100 million videos [84]. Problematically, the notion of relevance between two items of differing modalities is not semantic, but rather instance based. That is, captions are only deemed relevant to videos that they were collected with and vice-versa — regardless if similar videos/captions exist in the dataset.

This thesis explores the nature of video understanding for object interactions using open vocabulary labels. The thesis begins with closed-vocabulary action recognition that starts upon the path of expanding the vocabulary size, building upon ideas from information retrieval. The thesis then explores the fine-grained action retrieval task, as defined in chapter 5, which differs from both action recognition and general video retrieval.

1.1 Challenges and Contributions

In this thesis, four main challenges will be explored. These challenges will be briefly discussed here followed by the contributions of this thesis.

Challenge 1: Expanding the Vocabulary

Action recognition, being a classification problem, has seen datasets labelled with a closed vocabulary of both verbs and nouns to reduce the overlap between classes. For example, verbs such as *"put"* or *"place"* are semantically similar and therefore only one would be present in the annotations. Comparatively, spoken language — especially English — doesn't have this limitation, with different people offering different words when asked.

Collecting annotations with an expanded vocabulary can lead to issues, such as long-tailed distributions of classes, spelling errors or, ultimately, how to deal with the significant number of semantic overlaps. In the case of a model predicting *"put"* instead of *"place"* it shouldn't be penalised. This requires some external knowledge of how to relate the annotations in order for classification to be performed.

Challenge 2: Contextual Relationships

Previously, semantic relationships between verbs are the most common type of relationship that has been explored. Whether this is explicit semantic relationships, such as those found within WordNet [87], or those using co-occurrences as a measure of semantic similarity, as in Word2Vec [86] or GloVe [94], contextual relationships between verbs represent an under-explored area.

In certain contexts, two verbs can be interchangeable to describe an action, yet, for other contexts, they can be completely unrelated. For example, some doors are *"open[ed]"* by *"push[ing]"* whereas others are *"pull[ed]"*. *"Push"* and *"pull"* themselves are antonyms, and only contextually relevant with the verb *"open"* depending on the type of door being acted upon[1]. In this thesis, the term *contextual relationships* is defined to refer to these relationships between verbs which are only applicable in certain contexts.

Challenge 3: Scaling Relevance for an Open Vocabulary

Where action recognition datasets exclusively use closed vocabulary video labels, datasets used for video retrieval are all collected with natural language annotations, thus using an open vocabulary. However, in all video-retrieval works, videos are only relevant with the caption(s) associated with it, and vice-versa. If a caption of a different video is semantically valid, it is still considered irrelevant during both training and evaluation. For example, a video of someone folding origami in MSR-VTT ([144]) might have the caption *"a man doing an origami tutorial"*, a different origami video with the caption *"a man folds a piece of paper into origami"* would be considered as irrelevant as *"dancers are cheered on at a wedding"* as they are both captions for different videos.

Challenge 4: Training For the Unknown

When using an expanded vocabulary of verbs and nouns to describe videos, it can be impossible to have training examples for this expanded vocabulary, as well as all combinations of verbs and nouns, during training. For example, to include all of the ways that one can *"cut"* every type of vegetable in a dataset (*e.g.* chop, dice, julienne *etc.*) would require a huge number of training examples, contributing significantly to create a long-tailed distribution of actions. Because of this, it is likely that unseen classes will be present during testing. Known as Zero-Shot, this problem can be very challenging due to the difficulty of: Firstly, determining whether the test instance comes from a seen or unseen class and, secondly, being able to reason about unseen classes. The zero-shot task has not been explored for the task of fine-grained action retrieval or for video action recognition using verb-only representations.

1.1.1 Contributions

This thesis makes the following contributions:

- The notion of an expanded vocabulary of verbs for video action recognition is given

[1] As a more niche example, a video in the BEOID dataset ([22]) includes someone *"stepping on a treadmill pedal"* which also has the annotation *"place foot on pedal"* relating *"step on"* and *"place"* in (only) that context.

in chapter 3. An exploration into the usefulness of semantic knowledge bases (*e.g.* WordNet) for action recognition is also undertaken.

- Multiple-verb, verb-only labelling representations are introduced in chapter 4, presenting types of verbs, first introduced in linguistics, and analysing the contextual relationships between them for the tasks of action recognition and action retrieval.

- The problem of fine-grained action retrieval is proposed in chapter 5, where the aim is to retrieve semantically related items, be it visual or textual.

- A method for cross-modal embedding via disentanglement of different parts-of-speech is presented in chapter 5, showing its usefulness on both the fine-grained action retrieval task and the general video retrieval task.

- Chapter 6 includes experiments on both the multi-verb, verb-only representations (from chapter 4) and the cross-modal embedding method (from chapter 5) for zero-shot tasks.

1.2 Thesis Structure

This thesis has the following outline: Chapter 2 presents relevant background work to the tasks of both action recognition and information retrieval.

Chapter 3 first explores expansion of vocabulary sizes, purely for verbs, and presents a graph embedding method. In this chapter issues with using an expanded vocabulary of verbs are found.

Following this, contextual relationships between verbs are explored within chapter 4, via the collection of annotations. This results in experiments conducted for both action recognition and action retrieval, evaluated on the collected annotations. A downside to this verb-only approach exists though as, for large-scale datasets, this annotation process is expensive.

Chapter 5 focuses purely on the action retrieval task using the contextual clusterings of EPIC-Kitchens in order to bridge the gap between action recognition and video retrieval. A method is presented which creates separate embeddings for verbs and nouns, allowing for each to be independently modelled, before being combined later.

Next, Zero-shot applications of using an open vocabulary are presented in chapter 6 for the methods within chapters 4 and 5.

Finally, a conclusion of the work presented within this thesis will be given in chapter 7 as well as directions for future work.

Chapter 2

Background

This thesis attempts to discover how open vocabulary labels can be used for the tasks of video understanding, namely action recognition and action retrieval. This chapter provides a background to the related works upon which this thesis builds and is split accordingly:

Firstly, an introduction into the relevant Natural Language Processing and Linguistics techniques are given in section 2.1. Secondly, related works within the field of action recognition (including approaches used in later chapters of this thesis) are given in section 2.2. Finally, related works on the topic of information retrieval (covering methods employed for both retrieval in images and in videos) will be presented in section 2.3 and section 2.4 concludes the relevant works.

2.1 Natural Language Linguistics

This section will list different background techniques from natural language processing and/or linguistics which are relevant to this thesis. Specifically, WordNet [87] will be detailed in section 2.1.1, Word2Vec [86] in section 2.1.2, GloVe [94] in section 2.1.3 with the differences between the word encodings discussed in section 2.1.4. Finally, Manner verbs and Result verbs, being used in chapter 4, will also be introduced in section 2.1.5.

2.1.1 WordNet

WordNet [87] is an English Lexical database, created by lexicographers to include a number of semantic relationships between words. It is built up of many synsets which represent singular meanings. Each synset is assigned one or more lemmas, or words. For example, the synset which is defined by *"Cause to move by pulling"* has the lemmas of *"pull"* and *"draw"* associated with it, which can be thought of as synonyms to one another. WordNet includes 4 different types of synsets: Nouns, verbs, adjectives and adverbs. Synsets are normally labelled as ⟨word⟩.⟨type⟩.⟨num⟩. For example, *"put.v.2"* represents the second verb synset of *"put"*. The database includes different relationships defined between synsets with the relevant relationships being highlighted below:

- **Antonymy:** Whether two synsets have the opposite meaning to each other. For example, *"push.v.1"* is an antonym of *"pull.v.1"*.

- **Hyponymy:** Whether one synset is more general in meaning than the other, also called the IS-A relationship. For example, *"move"* is a hyponym of *"pull"*. Hypernymy is the reverse relationship, denoting a more specific meaning, where *"pull"* is a hypernym of *"move"*.

- **Troponymy:** Whether one synset is a more specific manner than the other. For example, *"tug"* is a troponym of *"pull"*.

The different synsets can be organised into hierarchies via hyponymy relations which is commonly used in computer vision to relate synsets together. ImageNet [25] is one such use case which groups noun synsets together. Nodes at the top of the hierarchy include those such as *"animal"*, *"instrument"*, *"plant" etc.* In this way, distances between synsets can be found using edge or node based similarity measures along the hierarchy of nouns. Similarly, a verb hierarchy can be constructed and used in the same way from the verb synsets.

The hierarchy, either verb or noun, can be used to determine a measure of similarity between two synsets, or nodes within the tree. A number of different approaches have been proposed to measure the semantic similarity between two synsets, largely falling into one of two categories: path-based methods [70, 143] or corpus-based methods [57, 73, 105].

Path-based methods, such as WuP presented by Wu and Palmer [143], generally involve finding the depth of the two synsets along with their least common subsumer (LCS). For example, the WuP distance between two synsets (represented by S_1 and S_2) can be found using the following formula:

$$WuP(S_1, S_2) = \frac{2 * depth(LCS(S_1, S_2))}{depth(S_1) + depth(S_2)} \tag{2.1}$$

where $depth(S)$ returns the height of the synset within the hierarchy and $LCS(S_1, S_2)$ returns the synset which is the least common subsumer (*i.e.* the lowest parent of both synsets related via the hyponymy relationship). This similarity measure returns a score between 0 and 1 with a high score representing that the two synsets are highly related.

Corpus-based methods instead use a corpus to relate two synsets. An example is the method proposed by Lin [73]. It uses the information content of two synsets to work out the similarity as below:

$$Lin(S_1, S_2, C) = \frac{2 * IC(LCS(S_1, S_2), C)}{IC(S_1, C) + IC(S_2, C)} \tag{2.2}$$

where C is the corpus being used and $IC(S, C)$ returns the information content of S using the following formula:

$$IC(S, C) = -\log(P(S|C)) \tag{2.3}$$

where $P(S|C)$ is the probability of finding the synset S in a given corpus C. Note that this can represent an expensive process in terms of annotation of senses for the chosen corpus.

It can be noted that both similarity measures presented here follow a similar structure, with WuP using the *depth* function and Lin using the *IC* function. Because of this, WuP can be useful if a corpus isn't available and/or only a hierarchy is present, whereas

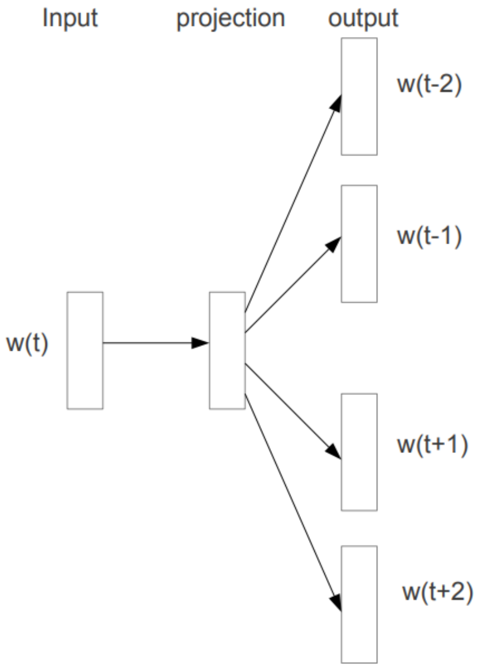


Figure 2.1: *Diagram of the Skip-Gram model first presented in [85]. For a given word at step t (represented by $w(t)$), the aim of the skip-gram model is to predict words around it in a sentence up until a set distance C, in this case $C = 2$ (figure from [85]).*

Lin can give more specific/contextual knowlege over the similarities between words but is dependent on the corpus.

2.1.2 Word2Vec

Word2Vec [86] is an unsupervised method for creating an embedding space of words. The authors build upon their previous work [85] which used a skip-gram model to learn vector representations of words from a corpus in an unsupervised manner.

Figure 2.1 shows an overview of the skip-gram model presented in [85]. A word in a sentence for a given time step t is projected using a log-linear classifier to predict words within a context window of a certain distance around the chosen word (denoted by C). For example, in the sentence *"put meat on ball of dough"* with $t = 4$ (w(4)= *"ball"*) and $C = 2$, the model will try to predict { *"meat"*, *"on"*, *"of"*, *"dough"*} for $\{w(t-2), w(t-1), w(t+1), w(t+2)\}$ respectively.

The skip-gram model was found to be more effective than the continuous bag-of-words model, also presented in [85], which can be thought of as the reverse of the skip-gram model (given a group of words around a missing word, can the missing word be predicted?). Regardless, both methods transform inputs via a projection matrix which is used to create the final embedding of each word.

The learned space provides a number of benefits: Words are represented as vectors in which their similarity[1] can be calculated using the cosine similarity. The space also allows for analogy tests to be performed using simple addition and subtraction of vectors. For example, *"King"* - *"man"* + *"woman"* gives *"queen"* as an output (word with highest similarity to the resulting vector). Finally, the choice of corpus allows for interesting zero-shot capabilities for visual tasks. As the learning is unsupervised and text corpora are a lot easier to collect than videos or images, it is likely that the vocabulary of words is much larger than present in vision datasets. With this (weak) knowledge of how words are related within the embedding reasoning can be performed on unseen visual classes.

2.1.3 GloVe

Global Vectors (GloVe) [**94**], is another unsupervised approach for learning an embedding for vectorised word representation, similar to Word2Vec (section 2.1.2). However, Word2Vec learns an embedding based on local co-occurrences using the context window, whereas GloVe instead uses global co-occurrences to construct the learnt embedding space.

In order to do this, given that the co-occurrences of words can be discovered within a corpus, probabilities of co-occurrences are first found, *i.e.* $P($*"ice"*$|$*"water"*$)$. Then, Pennington *et al.* present a method for using probability ratios to guide training. This can be seen in table 2.1 (reproduced from [94]). In this example, the co-occurrence probabilities of *"ice"* and *"steam"* can be seen with four other words: *"solid"*, *"gas"*, *"water"* and *"fashion"*. Note: *"water"* has the highest co-occurrence probability for both *"ice"* and *"steam"* with regards to the other words. As expected, the next highest co-occurrence probability for *"ice"* is *"solid"* and *"steam"* is *"gas"*. It should be noted though, that both $P($*"fashion"*$|$*"ice"*$)$ and $P($*"gas"*$|$*"ice"*$)$ have similar probabilities (same orders of magnitude) which is the same also for *"steam"* with *"solid"*.

[1] Whilst not strictly representing semantic similarity, this measure can still encode some semantic information.

Probability and Ratio	$k =$ *"solid"*	$k =$ *"gas"*	$k =$ *"water"*	$k =$ *"fashion"*
$P(k\|$ *"ice"*$)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|$ *"steam"*$)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|$ *"ice"*$)/P(k\|$ *"steam"*$)$	8.9	8.5×10^{-2}	1.36	0.96

Table 2.1: *Co-occurrence probabilities (first two rows) and ratio (bottom row) of "ice" and "steam" to four other words within a large corpus. Reproduced from [94].*

When the ratio between co-occurrences is investigated, a number of useful observations can be made. Firstly, if a word strongly co-occurs with either *"ice"* or *"steam"*, then the resulting ratio will be (respectively) very large/very small (*"ice"* is highly co-occurrent with *"solid"* giving a large value for the ratio whereas *"gas"* is highly co-occurrent with *"steam"* resulting in a very small value for the ratio). Secondly, if the word is equally relevant to the pair then a ratio of around 1 will be seen. Because of this, the ratio is better at determining the relevancy between words than looking at the base probabilities alone and, accordingly, Pennington *et al.* train a model of the following form:

$$F(w_i, w_j, \tilde{w}_k) = \frac{P(i|k)}{P(j|k)} \tag{2.4}$$

where w_i and w_j are word vectors of the ith and jth words in the corpus and $\tilde{w}_k$ is a separate context word vector. In this case, the right hand side of the expression represents the *"weakly supervised"* knowledge from the underlying corpus and F the generator which generates the vectors[2]. In their work, F is represented by a weighted least-squares regression model.

2.1.4 Word2Vec *vs.* GloVe

Both Word2Vec and GloVe represent unsupervised word embedding methods and are often compared. From the results within [94], Pennington *et al.* show that GloVe

[2] Note that the generated set of vectors, W, and the generated set of context vectors, $\tilde{W}$, are different, though the authors note that each set performs similarly. Indeed, for their experiments the final vector representation for the ith word is given by $w_i + \tilde{w}_i$ which they find to slightly outperform either w_i or $\tilde{w}_i$ alone.

performs better than the skip-gram model of Word2Vec on the word analogy task (*"a is to b as c is to _?"*). However, they note the difference in code setup between the two methods and state: *"We set any unspecified parameters to their default values, assuming that they are close to optimal, though we acknowledge that this simplification should be relaxed in a more thorough analysis."*

Regardless, this thesis focuses on the tasks of video understanding and, as such, performance in the word analogy task isn't necessarily a good indication, or even recommendation, for using one method over the other. In chapter 5, in which the word vectors are used as initial values to be transformed into a new visual-text embedding, the vectors are used only as input values (where discrimination between words is arguably the most important property). Because of this, the difference in performance between Word2Vec and GloVe was found to be marginal in chapter 5 and, due to simplicity and availability, Word2Vec was used.

2.1.5 Manner and Result Verbs

Verbs of Manner and Verbs of Result have been proposed by linguists in many previous works [6, 12, 35, 37, 45]. Specifically, the two definitions below summarise the different explanations of the related work above:

- **Verb of Manner**: A verb of manner will describe how an action is to be performed and won't include information about the final state. For example *pour* tells the actor to remove some substance from the object.

- **Verb of Result**: A verb of result will describe the end state of the object that the action will be applied to. *E.g. empty* tells the actor to empty the object of all substance[3].

Interestingly, Gropen *et. al.* [45] go even further by linking the verbs of manner to objects:

> "... the meaning of the verb pour specifies the particular manner in which a substance changes location — roughly, in a downward stream. For now it does not matter exactly how we characterize the manner in which a poured substance moves; what is crucial is that some particular manner of motion is specified in the meaning of the verb ...

[3] There are also different result verbs which could be used depending on the focal object. For example, an action could cause one object to be *emptied* to *fill* another.

> In contrast, the meaning of the verb fill specifies the particular way in which the ground is affected: a container must undergo a change of state from being not full to being full. Yet fill does not specify anything about the manner in which a substance is transferred ..."

This shows well the relationship between the two different sets of verbs and can be used as a good method to discriminate into which set a certain verb lies. Importantly, by explaining to an actor the action with only the verb of manner will leave them no goal or finish point, *how long do I need to keep pouring?* whereas supplying only the verb of result will leave the actor no tangible way of completing the goal, *How do I fill a container?*

Although not mentioning verbs of manner or result, Clark and Clark [20] go into detail about how nouns can be used as verbs which is applicable to this thesis through the use of tools as verbs — such as *scoop, spoon* or *spray.* Clark and Clark use the term instrumental verbs to describe these types of verbs. These are manner verbs in the contexts of action recognition dataset(s), which is expected to also be the case in other contexts as well. This is due to the nature of certain motions being tied to the instruments in their nominal role (for example, it would be odd to describe someone breaking an object with a spoon as *"to spoon the object"*, whereas for its main use case, *"to scoop an object"*, it is quite natural).

This thesis offers the first insight into how these complex linguistic meanings of verbs relate to the computer vision tasks of action recognition and action retrieval. However, the work presented in this thesis is only an initial investigation into how action verbs relate within the boundary of vision and language, leaving large scope for future work.

2.2 Action Recognition of Videos

Action Recognition of videos is a classification task, and can be thought of as analogous to object recognition for images. The aim is: For a video segment predict the class in which the video segment belongs to (the temporal extent of the action to be recognised is assumed to have been found or given by some oracle). Whilst action recognition can indeed be performed for images, certain actions, such as *"put down"* or *"pick up"*, can appear indistinguishable from one another due to the lack of temporal information.

With wearable devices becoming more common, a new perspective has been introduced, that of the egocentric perspective. Instead of a fixed camera which surveys a scene, as would be common for films or television, egocentric videos are recorded from the view of the participant performing the actions, giving a first-person view of the world. Camera's are generally mounted on the recorder's body, usually the head or chest.

More recently, the nature of granularity within action recognition has also been explored with several new datasets [23, 44, 46, 108, 116] being created with finer-grained action labels. Compared to older datasets in which the actions could describe multiple different events, these ensure that only a single action takes place in the video clip. For example, in CMU-MMAC [24, 130], the action *"put baking-pan [in oven]"* includes the actions of *"opening"* and *"closing"* the oven whereas in a dataset such as EPIC-Kitchens [23] the action would instead be labelled as three separate fine-grain actions.

Due to the shift in domain, approaches which tackle egocentric, third-person or fine-grained action recognition often use a differing set of low-level cues to recognise the action currently occurring. Accordingly, related works and datasets in third person action recognition will be presented in section 2.2.1 whereas section 2.2.2 will contain relevant works and datasets for those tackling the egocentric domain and, finally, section 2.2.3 introduces related works and datasets for fine-grained action recognition.

2.2.1 Third Person Action Recognition in Videos

After the success of image recognition, recognising actions in videos became an obvious extension, especially given the huge volume of videos being uploaded to sites such as YouTube [4] every day. The addition of the temporal dimension can help discrimination between actions, but it also represents a veritable challenge in how to perform video understanding.

Before the advent of deep learning, hand-crafted features were the supreme rulers. An early work [68] used Scale-Invariant Feature Transform (SIFT) [77] to track interest points throughout the video where local features are extracted around the interest points. The local features are then pooled with Bag-of-Words (BoW) [21] and an SVM is used for classification of actions. This, for many years, represented the standard pipeline to perform action recognition: Features or encoding methods changed and/or improved as well as different classifiers or optimisations were proposed.

Improving the action recognition Pipeline

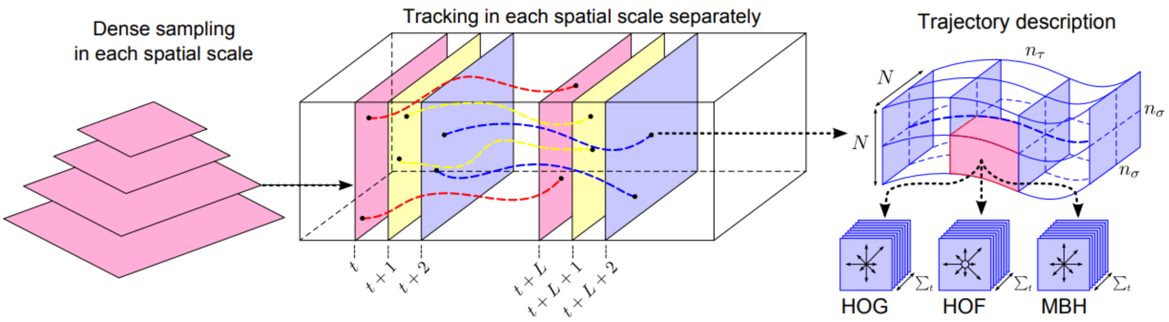


Figure 2.2: *For each spatial scale, multiple trajectories are sampled and tracked separately throughout the video. For each frame in the video, knowledge such as relative position as well as hand-crafted features (*e.g. *HoG, HoF or MBH) are extracted. Figure from [134].*

Notably, Wang *et al.* [134] were inspired by previous success of dense sampling in images and proposed a dense feature trajectory tracker, called dense trajectories. Feature trajectories combine the use of an interest-point tracker with a set of feature extractors such as Histogram of Gradients (HoG), Histogram of Flow (HoF) and/or Motion Boundary Histograms (MBH). An overview of the approach can be seen in figure 2.2 (from [134]). Compared to the previous approach of feature trajectories, which used Lucas and Kanade [78, 132] (KLT) or SIFT, they find their approach produces more trajectories with a higher overall quality. In this paper, the authors additionally propose using a motion flow field to track trajectories at multiple different scales to further the robustness of the tracker.

To perform experiments, the dense trajectory features are first encoded into a BoW representation [21] and then classified with a χ^2 SVM. As a baseline, the tracker from [132] is used with the same extracted features from each of the points as the dense trajectories features. On a variety of video classification datasets, including UCF Sports [106], dense trajectories were able to beat the baseline as well as give comparable or better performance than previous state-of-the-art performance showing the power of a dense representation.

An extension to dense trajectories, labelled as Improved Dense Trajectories (IDT), was proposed in [133]. In order to improve performance, camera motion is estimated and removed using a combination of SURF features [11] and optical flow to match keypoints across frames. A further addition, in the form of a human detector, is used to remove matches between keypoints within regions classified as a human body when considering global camera motion removal. The improved dense trajectories became the de-facto standard as baseline features to beat.

Focussed instead on the encoding stage of the pipeline, Perronnin *et al.* [97] improve on using the fisher kernel to encode visual features, originally proposed in [95]. Introduced as a counterpart to using BoW [21], the use of Fisher vectors allows for encoding of second order statistics within the encoding representation via a Guassian Mixture Model. However, when using Fisher vectors as presented in the seminal work, it had *"led to somewhat disappointing results — no better than [BoW][4]"*.

In order to improve the fisher vector encoding method, the authors propose 3 extensions. Firstly, by L2 normalising the feature descriptors, the fisher representation is able to reduce the background information of the scene allowing for more discriminative features for small objects. Secondly, a power normalisation is suggested to ensure that, in the case of a large dimensionality fisher vector encoding, the representation becomes less sparse. Finally, fisher vectors were sampled in a spatial pyramid to better *"take into account the rough geometry of a scene"* [69].

With the success of deep learning for image classification, Simonyan *et al.* [122] create a fisher layer as a general version of the standard fisher vector. The proposed layer can be added to and stacked within a CNN. The layer takes as input PCA-de-correlated features and learns a mixture of gaussians, similar to above. However, due to the large size required for the vectors, a linear transformation is learnt to reduce dimensionality of the fisher vectors[5]. A spatial pyramid is also used to learn the fisher vectors, so that each fisher vector only represents a portion of the image, and these are spatially stacked to get the final representation which is L2 normalised and de-correlated, using PCA once again, to allow for stacking of fisher layers. When compared to the improved fisher vector implementation [97], even a single fisher layer can lead to an increase in performance ($\sim 1\%$), whereas stacking the layers gives a large boost in performance ($\sim 5\%$) for object recognition.

Large-Scale Datasets

Early datasets for video action recognition, such as [56, 68, 75, 81, 106, 112], were small with only a handful of classes. With the IDT/FV/SVM pipeline providing a large jump in performance compared to other state-of-the-art approaches, the need for larger, more challenging datasets became clear. Two such datasets, HMDB-51 [65] and UCF101 [124] became the common test beds for this era, though others such as UCF50 [103] were also

[4] The paper used the term BOV in order to distinguish bag-of-visual-words from bag-of-words.

[5] With a Gaussian Mixture Model of size K and a feature size of d the resulting fisher vectors have a size of $2Kd$. This is (generally) projected down into the number of classes for their experiments.

introduced.

HMDB-51 [65] The Human Motion Database (HMDB) is a third person video dataset including 51 action classes with a minimum of 101 examples per class. At the time of its release, with 6,766 videos, HMDB represented the largest dataset for video action recognition. The dataset was collected manually, with students watching videos from various internet sources, such as YouTube [4]. The students then chose action clips where a single non-ambiguous action was present from the 51 chosen classes[6]. Additionally, videos were picked specifically to ensure a certain level of quality.

The 51 action classes can be grouped hierarchically, with the authors proposing 5, manually chosen, higher level categories consisting of: facial actions (4 classes); facial actions with object manipulation (3 classes); body movements (19 classes); body movements with object interaction (18 actions) and body movements for human interactions (7 actions).

They present a number of baselines for the dataset, showing the challenge of the dataset for hand-crafted features. Particularly, they note how removal of camera motion only leads to a small increase in performance for HoG/HoF features and, interestingly, a decrease in accuracy for C2 features [115]. They concluded that HMDB-51 represented *"a good place to start"* for furthering action recognition, highlighting that because state-of-the-art features (of the time) were unaffected by changes in camera position, camera motion or even object occlusion, it is likely that there was a large gap between recognising the low level and high level features required for successful action recognition.

UCF101 [124] Released the following year after HMDB-51, UCF101 pushed the size of action recognition datasets further, consisting of 27 hours of data made up of 13,320 videos grouped into 101 action classes. The dataset represented an extension of a previous dataset, UCF50 [103], adding 51 new action classes[7]. The videos are similar to HMDB-51 consisting solely of downloaded YouTube videos from the web.

Action classes are also grouped into 5 coarser grained action categories including: Human-object interaction (20 classes); Body-motion only (16 classes); Human-human interaction 5 classes); Playing musical instruments(10 classes) and Sports (50 classes). They note

[6] Example actions include *"diving"*, *"climb stairs"*, *"fencing"*, *"running"* or *"golf"*.

[7] Example new classes include *"Archery"*, *"Knitting"*, *"Long Jump" "Playing Flute"*.

in their baseline experiments that actions within the sports category achieve the highest accuracy which they posit as due to the distinctive actions and less-cluttered background scenes. In comparison, human-object interaction classes, with scenes in varied and cluttered living areas rife with object occlusion, scored noticeably lower.

The Deep Learning Takeover

The combination of larger, more challenging datasets and deep learning's continual success on image tasks, gave rise to deep learning approaches for action recognition. Initially, frame level features were extracted from CNNs trained on images and used in lieu of the (IDT) features in the action recognition pipeline. This naive approach has a drawback in that it doesn't allow for explicit temporal modelling between frames. Later, works developed to use CNNs in an end-to-end manner, completely replacing the standardised pipeline.

Karpathy *et al.* [59] focus on developing CNN approaches for video action recognition, inspired by the previous advances of using CNNs for image recognition. Additionally, they released the large-scale Sports-1M dataset which included 1 million YouTube videos weakly annotated with 487 different sports classes to train and test their model.

They note that *"Unlike images which can be cropped and re-scaled to a fixed size, videos vary widely in temporal extent and cannot be easily processed with a fixed-sized architecture"*. Therefore, to learn spatio-temporal features, their method takes as input stacks of the frames of the video. Furthermore, to improve the efficiency of their model, without a loss in accuracy, they propose two streams: Fovea and Context. The former uses a cropped high-resolution image as input whereas the latter uses a low-resolution image of the entire frame. In this way, their network can encode features from the entire frame of the video whilst using the bias of actions occurring in the centre of the frame to preserve high resolution details, which they note as necessary for action recognition. To train the CNN for videos, and to increase the robustness of the predictions, the authors sample 20 clips from each video which is propagated through each network with multiple crops and flips. The final predictions scores for each class are then averaged. This pre-processing step can be seen in many video action recognition methods which utilise CNNs.

Their experiments show the power of CNNs which, even with only having seen 20 clips as well as *"significant label noise"*, their method is able to beat the baseline feature histogram-based approach. They also find that a slow-fusion approach, which extends the base CNN to add temporal connections throughout the network, achieves best performance on the action recognition task.

In a similar fashion to above, Simonyan and Zisserman [120] propose the extension of deep convolutional neural networks from image representations towards video action recognition. Previous attempts had focused on stacks of RGB frames to learn both appearance and motion features, whereas this work proposed a two-stream network for both RGB and flow. The first stream takes in singular RGB images from a video and attempts to learn spatial features from the video. The second stream has stacks of flow frames for its input, allowing for the discovery of motion to be used for prediction of action classes. The two streams are fused via late-fusion (averaging the scores) or the use of a linear SVM on top of the features. In their experiments, using a fully connected layer to fuse scores resulted in networks overfitting.

They perform experiments on two datasets, HMDB-51 and UCF101, including a variety of different evaluations of both the spatial and temporal streams separately. Firstly, they note that by training on either dataset alone, both streams tend to overfit on the datasets — even though they were the largest available at the time of publication. To more effectively train their model, they perform multi-task learning across both datasets due to the difficulty in matching classes across the two datasets. Furthermore, the temporal stream was found to achieve best performance when using mean subtraction in addition to using bi-directional flow: including forward flow after the start frame and backwards flow before the start frame.

In regards to the fusion performance, the SVM performed moderately better over the averaged probability scores across both datasets. However, importantly, the different visual modalities were seen as complementary leading to the fusion model having an increase in accuracy over both spatial and temporal separately.

As a follow-up from the previous work, Feichtenhofer *et al.* [33] investigate different techniques of fusing information from the spatial and temporal streams, including via the use of three-dimensional convolution and pooling operations allowing for end-to-end training. Their base architecture follows that of [120], with a proposed set of fusion functions. Simple functions such as sum, max and concatenate are used as well as convolution and bi-linear fusion. Additionally, to fuse 3D information, two new methods are proposed along with standard 2D pooling[8]: 3D pooling as well as 3D convolution followed by 3D pooling. The other question the authors attempt to solve is that of

[8] In this case 2D pooling doesn't attempt to fuse temporal information at all and is a baseline.

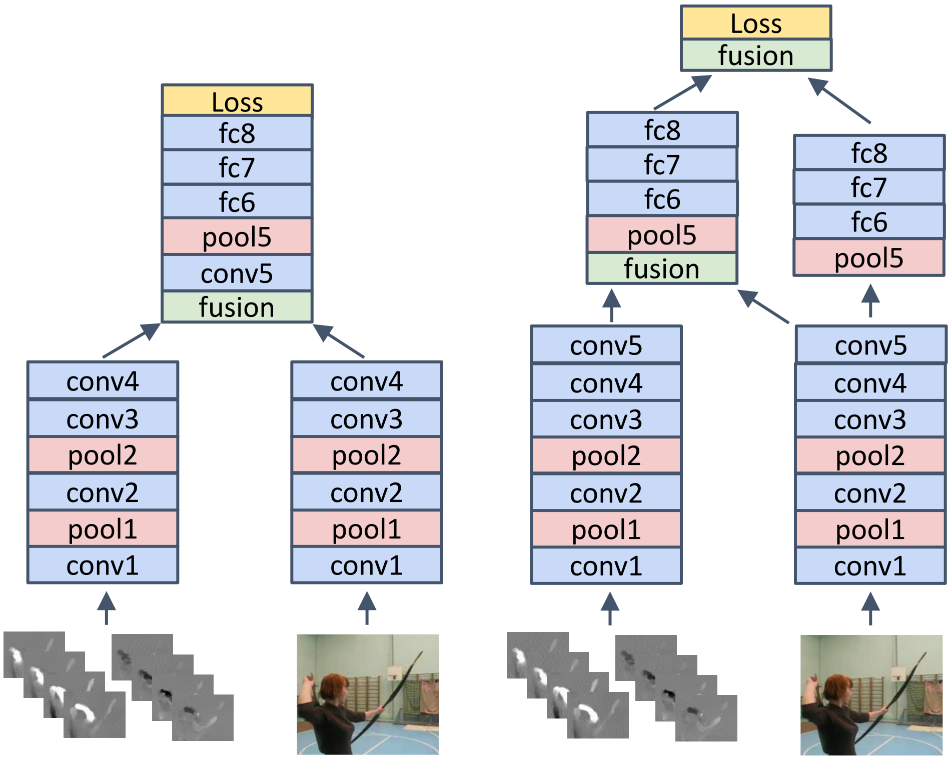


Figure 2.3: *Different architectures for performing two-stream fusion, based on the VGG-M [17] architecture. Left: both the spatial and temporal streams are fused resulting in a spatio-temporal stream. Right: One stream is fused into the other leaving a (in this example) spatial stream and a spatio-temporal stream which are later fused together. Note that the placement of the fusion layers isn't fixed in either case and can be done earlier or later within the networks. Figure from [33].*

"Where to fuse the networks?", and proposed two types of two-stream fusion network which can be seen in figure 2.3 (figure from [33]).

Again evaluated on both HMDB-51 and UCF101, the authors first test the method of fusion the networks finding that out of the three non-learned functions, sum achieves the highest recognition rate. Convolutional performs marginally better, however it is found to be comparable to the late fusion approach tested previously, but with almost 50% of the parameters. For *"where to fuse?"*, results show that earlier than the fifth convolutional layer results in detrimental performance. Additionally, when using two fusion layers (as in figure 2.3 right) a very small increase in accuracy can be seen, but at the cost of increasing the number of parameters by a factor or 2. When evaluating the temporal fusion method, they found that the combination of 3D convolution and 3D pooling boosts accuracy further than using 3D pooling alone. Finally, whilst they outperform conventional hand-crafted features (by around 1%) they find that the combination of their two-stream fusion model and IDT allows for a further 1% boost in performance.

Even Larger Large-Scale Datasets

3D convolution and end-to-end training of CNNs allowed for near saturated performance on both HMDB51 and UCF101 (though the former remains more difficult than the latter) through the use of pre-training. Additionally, CNNs were found to overfit on both datasets if used solely for training ([120]). As was the case when the two datasets were proposed, in order for video understanding and action recognition to be pushed further, the ever data-hungry CNNs would require much larger amount of training examples.

Kinetics [13, 14, 60] Kinetics was the answer to how can CNNs be trained for action recognition with more data and, on its release, was the largest action recognition dataset. The videos were collected from YouTube, but clips were collected from different YouTube videos to ensure more variation: UCF101, for example, includes multiple clips from the same video. Initially released with 400 action classes, the dataset was later extended to 600 [13] and, much more recently, to 700 [14] action classes.

The collection process for the dataset began with YouTube searches finding candidate videos for each of the chosen classes of which clips were extracted automatically using image classifiers. Amazon Mechanical Turk [1] (AMT) workers validate the presence of the action within the clip. Once each clip was ensured to contain the action, they were cleaned by removing duplicates (both automatically and manually) as well as removing videos *which had a high visual overlap with other classes.*

As part of the results, the authors discuss two biases that can be inherent within datasets: namely class imbalance of gender in addition to the bias of using image classifiers to automatically localise the clips. Through inspection, they note that 340 classes are gender neutral, *i.e.* not dominated by either gender, and heavily biased classes include examples such as *“shaving beard”* or *“filling eyebrows”*. However, classes which are imbalanced in such a way were found to still perform well on the less frequent gender. Other imbalances, such as age or race, were found to also have little bias. For baselines, three differing end-to-end approaches are considered: LSTM on top of a frame-based CNN, a two-stream fusion CNN (from [120] and expanded upon below) and a 3D CNN. Performance across all RGB baselines was similar, with the two-stream network achieving higher accuracies only when RGB and flow was used.

Going Against the Grain

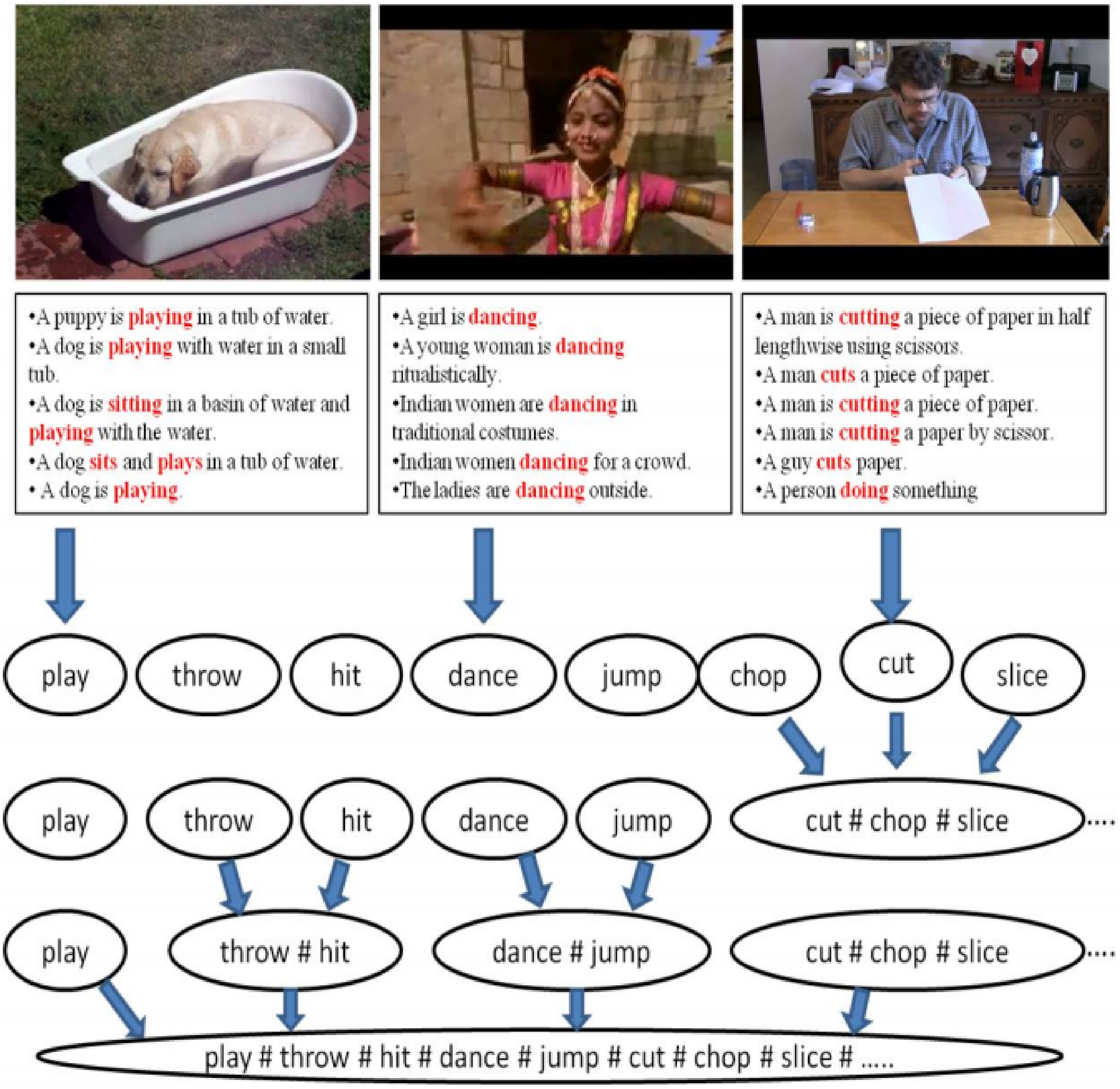
Previous works have focused purely on the task of multi-class, single-label action recognition where classes are represented by a single word (*e.g. "Swimming"*) or verb-noun phrase (*e.g. "Kick Ball"*). Additionally, approaches solved this problem using a spatio-temporal interest point approach. However, the works presented below instead expand the notion of video understanding by treating it as a multi-label problem, using natural language or by focussing on objects.

Motwani and Mooney [91] combine weak supervision for action classes in the form of natural language descriptions, object recognition methods and priors from text corpora, to improve action recognition performance. To automatically discover actions from long form natural language annotations they first parse and stem the words. The verbs are then extracted and the most common verb is chosen per video. To further build the verb hierarchy, WordNet [87] is used. Firstly, the most common senses for each verb are chosen before a combination of path-based similarities (*e.g.* WuP [143]) and corpus-based similarities (*e.g.* Lin [73]) are used to cluster the verbs and create a hierarchy. For their experiments, a threshold is chosen so as to *"create meaningful [action] classes"*. Examples of this approach can be seen in figure 2.4 (from [91]).

These labels are used as supervision to train a decision tree classifier that is based on HoG and HoF features extracted from spatio-temporal interest points. These features are augmented with an off-the-shelf object detector along with priors from a text corpus. From their results, the object detector and the decision tree classifier perform similarly for the task of action recognition but the combination of both gives a significant boost to accuracy. Similarly, the prior knowledge of co-occurring actions also gave increased performance.

Khamis and Davis [61] introduce the notion of action recognition being a multi-label problem where previous approaches treated it as a single label, multi-class problem. For example, whilst a person might be *"waiting"* in a queue they are also *"standing"* and could also be *"talking"*. To solve this, they re-annotate the UCL-Courtyard dataset[9] with multiple labels. They extract feature trajectories using KLT and HoG/HoF before build-

[9] This dataset includes two viewpoints of a courtyard on a university campus and many actors performing 10 different actions.



- A puppy is playing in a tub of water.
- A dog is playing with water in a small tub.
- A dog is sitting in a basin of water and playing with the water.
- A dog sits and plays in a tub of water.
- A dog is playing.

- A girl is dancing.
- A young woman is dancing ritualistically.
- Indian women are dancing in traditional costumes.
- Indian women dancing for a crowd.
- The ladies are dancing outside.

- A man is cutting a piece of paper in half lengthwise using scissors.
- A man cuts a piece of paper.
- A man is cutting a piece of paper.
- A man is cutting a paper by scissor.
- A guy cuts paper.
- A person doing something

Figure 2.4: *Automatic discovery of verbs from natural language descriptions of videos. Most frequent verbs are chosen before a hierarchy is created using WordNet. Figure from [91].*

ing an extension of a one-vs-all SVM approach via learning both the prior probabilities of each class and the standard weights using an alternating optimisation problem.

Their final label correlation matrix, which is learned during training, notes that actions such as *"walk[ing]"* and *"talk[ing]"* are positively correlated, whereas *"eat[ing]"* and *"bi[cycling]"* are negatively correlated. Their approach of learning the action priors in this way beats the baseline 1-vs.-all SVM, even with a small dataset in which only 56.9% of videos have two or more occurring at once (and only 4.9% have three or more).

Jain *et al.* [54] use objects to encode actions, diverging from the standard pipeline which used spatio-temporal features extracted from trajectories. Their method sees the training of a 15,000 object class classifier (all classes from ImageNet [25] which, at the time, included $>$ 100 examples). Each action is then encoded as a mixture of different objects: For example, *"Typing"* may be composed of *"computer-keyboard"*, *"keypad"*, *"computer" etc.* Interestingly, some objects for an action denote active objects (those that are interacted with as part of the action) whereas others are seen in the background which can cause issues if multiple actions share the same or similar backgrounds.

They test four different scenarios on four different datasets including UCF101 and HMDB-51. Firstly, somewhat unsurprisingly, datasets which include actors interacting with objects(such as UCF101) see a boost in performance when object information is combined with motion information (using improved dense trajectories [133]). They also show that actions have an object preference, in which each action class was tested with only a subset of all relevant objects. Results showed that as little as 11 objects gave the best accuracy. Objects are also shown to generalise across datasets between UCF101 and HMDB-51 leading to increases in accuracy on HMDB-51 when UCF101 objects are learnt. Finally, they deduce that the addition of objects can help boost performance to improve the state-of-the-art results which, at the time, used motion information over object information.

As an extension to the previous work, Jain *et al.* [53] create a semantic word embedding of objects for the task of zero-shot action recognition (where new classes are presented at test time, see chapter 6 for more information). Their method uses ImageNet [25] to learn representations of objects from images. They then perform domain transfer to translate object semantic information into the zero-shot action classes at test time. Compared to other embedding approaches, in this work they wish to embed object information near (relevant) action information and, appropriately, use the skip-gram model from [85] and fisher vectors [111] to embed the multi-word descriptions into the embedding space.

Concluding Remarks

Video Action recognition has undergone a large number of changes from its inception. With hand-crafted features, the set-up of the standard pipeline which saw features, extracted from trajectories or around spatio-temporal interest points, encoded into a discriminative representation for an SVM to classify the action. Larger datasets such as UCF101 and HMDB provided a new challenge requiring improvements of the pipeline, but it was the use of end-to-end CNNs which saw its end. Deep learning action recognition approaches focus on learning both spatial and temporal information, though of course differ on specifics, but were found to overfit on datasets of the time. This led to the creation of much larger and challenging datasets such as Kinetics.

Action recognition has also seen the rise of a number of sub-problems, with some approaches focussing on expanding the small/closed vocabularies or even treating action recognition as a multi-label problem. Finally, the use of object features for action recognition have also been explored, allowing for zero-shot recognition and explicit encodings of actions.

2.2.2 Egocentric Action Recognition in Videos

Egocentric video refers to any video in which the camera has been mounted upon the participant's body. This is usually either the head or the chest, giving a very personal view of the world. Compared to third person video, both actions and objects can take up a larger portion of the frame. However, the egocentric domain can be particularly challenging. Head (and body) motion can cause large amounts of global motion within the scene, hands can obscure task relevant objects as well as sometimes actions occur off-screen, *e.g.* if a participant closes a cupboard door behind them whilst looking at something else.

The egocentric domain, and the accompanying datasets, represent an important challenge for the field of computer vision as they depict how humans interact with the world as well as becoming a model for how a robot might perform in a similar environment. This has many benefits for scenarios such as care of patients (such as those with alzheimers) or can be used as a supervisory tool/checking tool for robots to ensure that their interactions are correct/successful. Because of this, many tasks exist in the egocentric domain including video summarisation, activity recognition or, pertinent to this thesis, the task of action recognition.

This section will first present details of two Egocentric datasets which will be used as test beds in chapters 3 and 4 of this thesis. Next, related works of egocentric action recognition will be presented.

BEOID [22] and EPIC-Kitchens [23] As part of the collection/annotation of both datasets is a contribution of this thesis, information of each will be presented within the relevant chapters, namely chapters 3 and 4 for BEOID and chapter 5.

CMU-MMAC The CMU-MMAC dataset, presented in [125] and [24], is a multi-modal activity dataset — recorded with IMU data, multiple static microphones, static cameras for different views and a head mounted camera for an egocentric view of the scene. The dataset includes video recordings from 39 different participants performing 5 different recipes: Brownie, Eggs, Pizza, Salad and Sandwich.

The dataset only includes temporal, action-level annotations (from [130]), for the brownie task which included 13 different verbs and 19 nouns constituting 29 different action classes with an average length of 8.7s per video. In addition, actions were labelled such that there was no background class, *i.e.* every frame is labelled with an action class, leading to long actions with inconsistent start/end points.

The original work uses two methods to perform the task of action classification, a supervised Hidden Markov Model, or HMM, and K-Nearest Neighbours (K-NN), with $k = 3$ in this case. Gist features [93], extracted per frame, were used as features as the authors noted: *"many actions are performed while looking at a somewhat constant background"*. They found that the combination of IMU data and visual features proved fruitful, giving benefits of around 2% over using each modality separately for the HMM. The K-NN classifier was the best performing model in which the combination of both visual and IMU data gave an overall acuracy of 57.8% (compared to the 12.3% of the HMM). It is interesting to note that using IMU data alone achieves 56.8% compared to the visual features accuracy of 48.6%.

GTEA Gaze+ The GTEA Gaze+ dataset was introduced within [31], in which a large dataset was collected consisting of 5 participants performing 7 different activities within the kitchen (namely: American Breakfast, Pizza, Afternoon Snack, Greek Salad, Pasta Salad, Turkey Sandwich and Cheese Burger). For each of the different tasks, participants were given the same set of instructions to follow and all sequences were collected within the same kitchen. Additionally, gaze was also recorded for each participant using an eye-tracker. The actions were labelled using a closed vocabulary of 25 different verbs and 45 different nouns. The videos were rather short with an average length of 2.0s.

Motion, Gaze, and Hands

Compared with videos shot from a third person viewpoint, egocentric videos contain three large sources of information: Motion, in the form of head motion and action motion (which, comparatively, takes up a larger portion of the frame), hands, which can be used to localise objects/actions, and gaze information (if the videos were recorded with a gaze device).

In an early work, Sundaram and Mayol-Cuevas [129] concentrate on removing background noise and extracting hand/arm motion and background scene detection. They extract Histogram of Gradient (HoG) features across frames which are grouped along the temporal dimension. A graph is constructed out of random groups of gradients which is treated as the representation for an action. New videos can be classified at test time via their distances to action representations for each class.

To accompany their graph-based approach, the authors additionally use a Simultaneous Localisation and Mapping (SLAM) map which is built offline and employed during testing for re-localisation. The localisation adds a small increase in action recognition accuracy but also has the benefit of reducing the computation time required per frame of video.

In their work, Li *et al.* [72] state the importance of what they consider as mid-level egocentric cues: namely head/hand motion, gaze and hand pose. They combine features extracted to describe these cues along with the standard low level cues such as flow and object features using concatenation of Improved Fisher Vectors (IFV) [97]. A linear SVM is then trained on top of the representation to classify individual videos. Figure 2.5 shows the different features the authors use for action recognition (from [72]).

Experiments are conducted on GTEA Gaze+, among other egocentric datasets, along with a full ablation of different combinations of the different mid and low level cues that the authors described. Overall, they find that the combination of object, motion and egocentric cues outperformed previous state-of-the-art methods. However, the egocentric features, by themselves, perform poorly and give only a slight boost to accuracy when combined with the other low-level cues. Overall, for the GTEA Gaze+ dataset, object features were found to be the most important, likely due to how all videos were captured in the same environment with the same objects. The object features were also being unaffected by varying the method that found the attention point (either by localising hands or via gaze).

Whilst focusing on hand detection in egocentric video, Kumar *et al.* [66] also show the

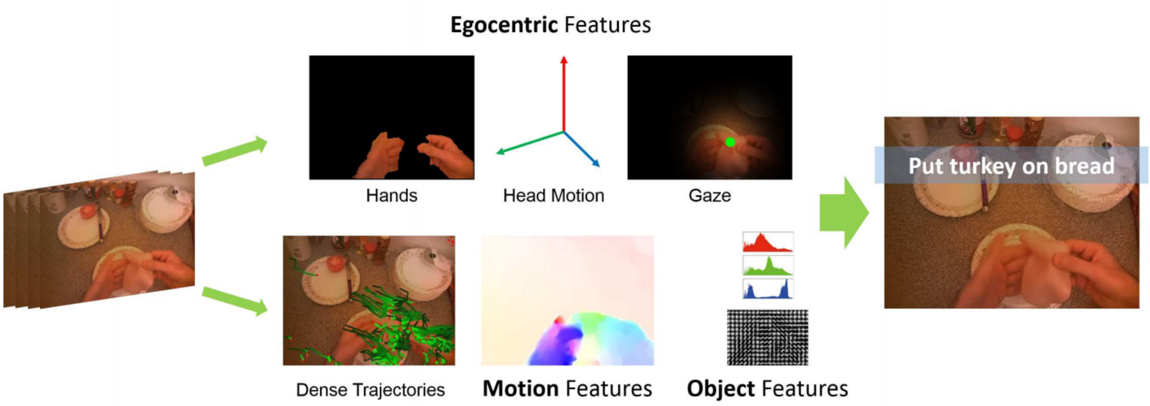


Figure 2.5: *Egocentric cues, such as hands, head motion or gaze are combined with low-level cues such as motion (via dense trajectories with motion compensation) and object features to perform action recognition. Figure from [72].*

benefits of such a method for action recognition. Their unsupervised hand detection first directs the user to perform a calibration technique that exhibits both the palm and back of the hand to the camera. Using this, thresholded optical flow features are extracted from the input frames and a region growing algorithm is used to segment the hand from the other elements in the scene. A Gaussian Mixture Model (GMM) is trained on the calibration frames to build a hand detector for subsequent parts of the video.

In order to recognise actions, the hand detector is used to locate hands within test videos. Dense trajectories are extracted around the hand mask, sampled using a 2D Gaussian function, and a BoW representation is computed. Classification is performed using a χ^2 Support Vector Machine (SVM). They show that their method is faster than competing methods whilst still achieving a similar level of accuracy without the use of a gaze tracker.

Much Ado About Objects

Whereas in third person action recognition objects were often avoided in preference of spatio-temporal interest points, for egocentric vision the opposite can be seen. The first-person view of the world means that objects and hands often occur in frame at much higher scales than in third person video, making them a natural choice for use in action recognition.

The works of Fathi *et al.* [28, 29, 31], were targetted solely on the egocentric domain and action recognition. Initially, in [30], the authors focused purely on object detection in egocentric videos by using segmentation, calculated using colour histograms to separate foreground/background, to find hands and objects within the scene.

The object and hands detector then features in the method of [29], which takes initial estimates of objects, hands and the background to predict action and activity labels. Once the activity labels are predicted, the whole pipeline is further refined in a backwards manner, *i.e.* activity labels correct action labels and action labels are used to correct object segments *etc.* By incorporating all three levels of the hierarchy in their method, the reasoning for each benefits greatly, allowing for erroneous predictions to be corrected.

To further understand egocentric actions, the authors then focus on gaze, in [31] — an interesting aspect only available in the egocentric domain. Their method jointly estimates the action taking place as well as the relevant sequence of gaze locations that the participant was looking at. They construct the hypothesis that the most informative features for the currently occurring action lie around the point the participant is looking at. A method is built by combining three sets of features combining object-based features, appearance based features and future manipulation features[10]. For inference, an SVM classifier is trained per action class (in a one vs. all manner) which is built upon a Hidden Markov Model (HMM) that models the gaze path.

The results show that gaze can be an important tool in correctly predicting the action, giving large increases in performance over using saliency alone. Similarly, using an oracle for gaze results in a significant boost in action prediction. Learning to predict both action and gaze simultaneously also leads to an increase in action prediction, albeit lower than when using the oracle.

Finally, Fathi and Rehg [28] highlight another facet of the egocentric domain which makes it particularly challenging: state changes of objects. Given the first person view, and a plethora of object interactions, egocentric actions contain the potential for fine-grained action labelling that distinguishes between state changes of the same object. For example, when opening and closing a coffee jar in a third person dataset the state change (the presence of a lid) would represent a small part of the video and could very well be occluded by the subject. From an egocentric point of view, these state changes are more likely to be visible in frame and represent a larger portion of the frame.

By modelling the state changes, the authors proposed a method that discovers changed

[10] Inspired by psychology literature which states that gaze precedes hand movement.

regions within the scene. By clustering these changed regions, actions can be predicted via the closest cluster to an unknown state change, leading to a more meaningful representation for human understanding (their model gives a high weight to an open jar at the start of the action and the closed lid at the end).

Ishihara *et al.* [52] focus on hand-object interactions within egocentric videos, stating the difficulty of the task due to *"constant motion, cluttered backgrounds, and sudden changes of scenery"*. In their work they note how it can be difficult to distinguish actions in which the same motion is present, but the hand is present in a different configuration. Accordingly, they proposed to use both dense local features, via dense trajectories [134], in addition to global hand shape features (using HoG and PCA) to recognise actions. They perform experiments on four different datasets, including CMU-MMAC and GTEA Gaze+, outperforming previous state-of-the-art via their combination of local and global features.

Egocentric videos from datasets often are cluttered, with scenes containing a large number of objects in view. However, object interactions are usually only applied to one or two active objects (*i.e.* those being interacted with). As such, a lot of works single out these active objects and/or explore the relations between them.

Focused on learning object interactions — specifically with active objects which they define as task-relevant — Damen *et al.* [22] presented an unsupervised approach for video guidance. Their approach, titled *"You-Do, I-Learn"*, takes as input multiple egocentric views of the scene from different operators along with gaze information. The distinction between fast eye transitions and fixations allows for images to be extracted when actions with task relevant objects are occurring. Histogram of Gradients (HoG) are extracted as features and represented via Bag-of-Words (BoW). Spectral clustering is then used for the unsupervised discovery of task-relevant objects.

As each active object can have multiple modes of interaction, Histograms of Flow (HoF) features are first embedded into a BoW representation before being combined in a pyramidal manner. The same clustering technique for task-relevant object discovery is also used to discover the different modes of interaction. Results are presented on the BEOID dataset (which is introduced in the same paper, see chapter 3 and 4 for more details) where 95% of objects can be discovered using gaze attention.

In their work, Pirsiavash and Ramanan [101] proposed the Activities of Daily Living (ADL) dataset. It included 1 million frames and 10 hours of video. Compared to

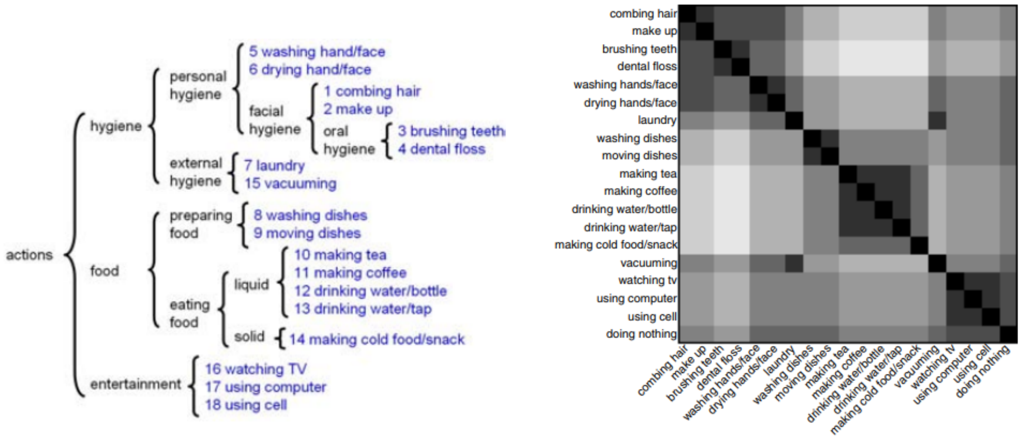


Figure 2.6: *Left: The manually created taxonomy of the ADL dataset. Actions are grouped via differing hierarchies of specificity, similar to the hypernym hierarchy within WordNet. Right: The distance between actions according to the hierarchy, calculated using a path-based metric. Figure from [101].*

other egocentric datasets available, ADL was unscripted[11] with participants recording in different homes giving a much more realistic and challenging dataset. Additionally, participants recorded their videos using a chest-mounted camera, giving a different view of the world compared to the head-mounted datasets (head motion, for example, is not present, but objects being interacted with are more likely to be obscured by hands). The authors also present a taxonomy of the actions that can be found within their dataset, which can be seen in figure 2.6 (from [101]). By using a path-based distance metric, similar to WuP [143], the distances between actions can be seen and mispredictions between classes can be penalised accordingly. Note how classes are relatively coarse-grained in addition to being non-overlapping.

Their proposed baseline for the dataset uses the notion between *"active"* objects (currently being used in the action) and *"passive"* objects (representing a background object), putting forward the notion that objects in their active state look different than the same objects in their passive state. Accordingly, they train an object classifier to detect only active objects using a subset of training images (including the spatial bias for locations of objects which are active compared to when they are passive). A temporal pyramid, which splits the frame into 2^i bins for the *ith* level, is used to accumulate the object features per frame and a linear SVM is trained on top of the concatenated features.

[11] Participants were given tasks to complete, but the freedom to do them how they wished and in any order.

For the task of object detection, they note that detectors trained on ImageNet [25] perform very poorly on ADL due to the large domain shift between the two datasets, and thus train models on their own dataset. Compared to the baseline methods, which use either spatio-temporal interest points or only object features, their active object based method achieves a large gain in performance. Additionally, by using the pyramidal structure over a flat BoW model, increases in action recognition accuracy can be seen across all methods, including when the proposed taxonomy of actions is taken into account. Interestingly, they also present results when the object detectors are *"idealised"*, *i.e.* achieve 100% detections, along with a oracle which determines active objects, a veritable boost in performance can be seen. They surmise that, for ADL at least, knowledge of the objects being interacted with is important to perform action recognition.

McCandless and Grauman [82] also focus on detection of objects to perform action recognition in the egocentric domain. Their approach first detects all objects within the scene regardless of if they are active or passive (similar to [101] above). Compared to the previous approach which only trains active object classifiers, they train active and passive object classifiers for each object available in the scene. A space-time pyramid histogram is then used to aggregate the objects over multiple spatial scales in each frame over time. In contrast to other previous work, the bins of the histograms are chosen randomly, with an object prior (given by the location of active objects in the frame which is calculated prior to training). A multi-class SVM is used as a classifier for each spatial-time pyramid. Due to the high numbers of randomised pyramids that are generated during training, a boosting approach is used to create a strong classifier.

They perform their experiments on the Activities in Daily Living (ADL) dataset [101] noting that their boosted approach beats the previous temporal window-based approach. The object prior, *i.e.* choosing bin locations based on active objects, leads to increased accuracy with a lower number of pyramids, increasing performance over the random sampling method. However, their method understandably struggles in distinguishing between actions in which the same objects are present (for example *"making tea"* and *"making coffee"*).

The previous works have noted the importance of object recognition for action recognition tasks within egocentric video. In their work, Ren and Gu [104] develop a method which attempts to reduce detections of objects within the background of the scene (*i.e.* the passive objects from [101]). Their method uses a combination of optical flow and knowledge of object priors to segment out the background from the scene leaving only the hands and object(s) being interacted with.

Beyond Short-Term Modelling

Most approaches discussed above are applied towards short-term video understanding and/or don't explicitly model temporal information. As most egocentric datasets contain actions which are more fine-grained actions than their third-person counterparts, approaches generally focus on objects. However, when representing coarser grained actions, such as those in CMU-MMAC, stricter modelling of temporal relationships can provide useful benefits.

Lade *et al.* [67] approach the task of understanding cooking activities via the use of the underlying actions. Their method uses knowledge of the activity in addition to previous actions to predict the most probable next action. First, the authors cluster actions for two of the activities in CMU-MMAC [24, 130], highlighting how some actions such as *"take spatula"* can occur at any time and so cannot be grouped with others and later used to predict the next action. Secondly, they build two models: A hidden markov chain to model the actions in a hierarchical fashion and a set of transition matrices which determine object usage. The combination of these models allow for the proposed method to track actions that have occurred using the knowledge of action and object ordering to predict the next action.

Ryoo *et al.* [110] propose a method of temporally pooling video features using a *"histogram of time"*. They propose the pooled time series representation which aims to capture both short and long term changes in high-dimensional features. Their framework allows for either CNN or hand-crafted features, which are extracted per video frame, and aggregated into time series per element (for example, they extract CNN features of length $4,096$ and so create $4,096$ different time series). Each time series is then pooled temporally using a pyramid of k histograms creating a $n \times k$ length final representation where n is the size of the initial features. Classification is performed on the final representation using a χ^2 SVM.

They perform experiments on two different egocentric datasets comparing their proposed representation with both Bag of Words and Improved Fisher Vectors. They find that the time series representation is able to consistently outperform the baselines across both datasets regardless of the features being used (either from various pre-trained CNNs or hand crafted features such as motion boundary histograms, MBH, or Histogram of Flow, HOF) with CNN features, particularly those from Overfeat [114], outperforming classical hand-crafted features. Additionally, the temporal pyramid was tested on top of all three representations giving the highest benefit to the proposed representation.

Granularity	Action Labels		
Level 1	Manipulation	Non-Manipulation	
Level 2	One hand	Walk	Talk
	Two hands	Stairs	Seating
	Pick-up	Stand	Screen
	Others		

Table 2.2: *Hierarchy of action labels used in [89] for their action recognition task. A binary classifier can be used to determine whether an action involves object manipulation or not with secondary multi-class classifiers used to predict level 2 actions. Table from [89].*

Focussing on coarser-grained actions, Moghimi *et al.* [89] use the distinction between manipulation and non-manipulation actions to split actions into a hierarchy which can be seen in table 2.2 (table from [89]). In their work they experiment on the difference between global image features, such as GIST [93], compared with pre-trained CNN features (pre-trained from ImageNet [25]). Both methods feature an SVM which classifies the actions. Skin segmentation features and depth information were also used to support the features, with the latter being used to better segment the hands — *i.e.* ignoring *skin-like* pixels which are too deep within the scene.

Their results show the power of CNN-based representations which are able to outperform the global-image based representations, especially on manipulation actions. However, for the distinction between manipulation and non-manipulation, the authors find that a simple skin detection method is able to achieve comparable results on this coarser-grained task.

Deep Learning Resurgent

As seen with third person action recognition, the success of deep learning for images and objects was quickly investigated and applied for egocentric action recognition. Given the difference in domain, deep learning approaches for egocentric videos tended to build upon previous works and specifically build objects into their architecture instead of relying on an RGB stream to specifically model this information (as in [120] or [33])

Ma *et al.* [79] were among the first to apply deep learning for the task of egocentric action recognition. They argue that an action can be decomposed into two different observations: That of appearance, concerning hands and objects, as well as that of motion, via hand movement and ego-motion. Because of this, they present a method

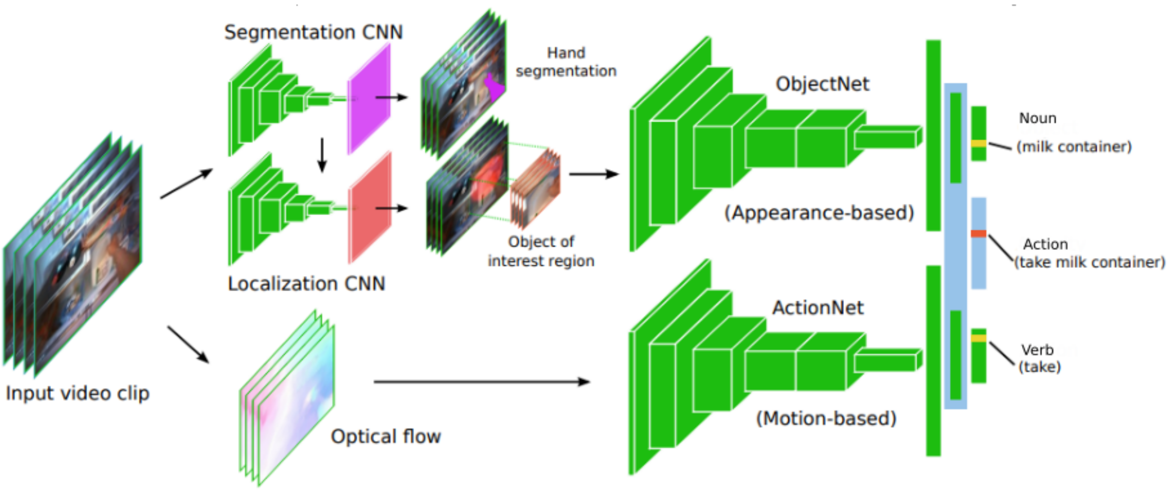


Figure 2.7: *Overview of Ma* et al.*'s method for egocentric action recognition. Figure modified from [79].*

which includes two streams predicting a noun and a verb which are fused to predict the action.

Figure 2.7 shows an overview of their method which includes two separate streams for object prediction and verb prediction. Two CNNs are used to segment and localise hands and objects respectively which form the input to the object stream. These networks are trained separately from the rest of the method and are necessary due to the unlikelihood that (relevant) objects appear in the centre of the frame. The noun and verb predictions are calculated through the use of a softmax layer at the head of the noun and verb streams respectively. In order to predict the action of the sequence the heads of each network are fused with a final fully-connected layer which allows learning of co-occurrences within the data (*e.g.* *"cut"* is not likely to be predicted with *"tea"*).

They perform experiments on three different egocentric datasets, including GTEA Gaze+. The proposed method outperforms all other approaches for the task of egocentric action recognition as well as both verb and object recognition. Note that the baselines weren't learned end-to-end, use hand-crafted features or are unspecialised to the egocentric domain. They also show the importance of training end-to-end with their method seeing a large drop in accuracy, compared to other two-stream fusion approaches, when an SVM is learned on top of the CNN features.

Singh *et al.* [123] propose a three stream CNN architecture for egocentric action recognition, with a trained end-to-end egocentric stream. They note the importance of egocentric cues such as hands and removal of head motion from previous works and so a single stream of their network is focused solely on learning these cues via the input of hand masks, camera motion and saliency maps. The other streams consist of the spatial and temporal streams from [120] which they use as feature extractors. An SVM trained on the extracted features is fused with the output of the egocentric stream via a learnt weighting to create the final class prediction scores.

Importantly, they note that the egocentric datasets they perform experiments on are too small to train from scratch, and so initially train on the interactive museum dataset [10]. Additionally, whilst the egocentric cues were able to achieve state-of-the-art results on the various egocentric datasets, by combining the two-stream network features another large boost in accuracy can be seen.

Concluding Remarks

In stark comparison to third person action recognition, egocentric action recognition techniques have placed a large importance of object understanding. Other egocentric cues such as head/scene motion, gaze and hands have also been used to great effect and deep learning approaches have specifically encoded this information within their architecture to improve results for the egocentric domain.

However, two interesting facets of this domain still remain largely underexplored, namely open vocabulary and verb modelling. Egocentric action recognition approaches have assumed action recognition to be a single-label classification problem with non-overlapping verb-noun classes (*i.e.* the set of verbs/nouns that are chosen that no two are semantically similar). This thesis aims to explore these two areas.

2.2.3 Fine-Grained Action Recognition

Fine-grained action recognition can be thought of as a challenging extension to action recognition tasks that were presented in the previous two sections. To differentiate, fine-grained action recognition includes shorter actions with labels which are highly specialised as to what is occurring. Additionally, the temporal extent of the action will include only what is labelled, not any other sub-tasks that could be required. For example, instead of simply *"place meat"*, the action might be labelled as *"place meat on ball of dough"*, requiring a successful method to reason not only what the action and the primary object the user is interacting with, but to also include knowledge of the surrounding scene in its prediction. This section first presents three datasets that have been released upon which fine-grained action recognition is performed before including relevant works which attempt this challenging task.

MPII Cooking 2 [**108**] Rohrbach *et al.* created the MPII Cooking 2 dataset to further research on the challenging task of *"detecting fine-grained activities and understanding how they are combined into composite activities"*. The dataset includes 30 subjects recording 273 videos from a fixed camera perspective. In total, there are 14,105 actions making up 67 different fine-grained classes. However, they note that, although participants were given dishes to prepare for the cooking tasks, they were free to prepare it in any way they wished. In order to provide baselines for the dataset, the authors explore the use of human pose and hand detection to boost performance compared to other leading approaches such as Dense Trajectories. Their results show the importance of trajectories sampled around hands which give a veritable boost when combined with dense trajectories.

Something-Something [**44, 80**] The Something-Something Dataset was introduced as a fine-grained action dataset where videos were labelled with templates which didn't specify the object as part of the action. For example, the class *"picking* something *up"* could refer to *"picking shoe up"* or *"picking phone up"* and videos including both actions would be assigned the same class. In its second release, dubbed v2 [80], videos were also annotated with their relevant objects[12]. Compared to datasets before it, Something-Something included both coarse and fine-grain action classes where the latter are grouped together. *E.g.* the coarse-grained class *"Putting* something *somewhere"* has the corresponding fine-grained action classes *"Pretending to put* something *on a surface"* and *"Putting* something *on a surface"*. When presenting their results, they find that by training on fine-grained classes provides better action recognition accuracy on coarse-grained features compared to training for coarse-grained classes alone.

Charades [**116, 119**] The authors of charades note how most actions during daily lives are typically unexciting. Compared to other datasets which include sports or cooking, everyday actions such as *"searching for keys"* don't occur often in movies, TV or on YouTube — of which most large-scale datasets use to collect their data. To record these

[12] The second release also doubled the size of the dataset, increased video resolution and reduced label noise.

day-to-day actions, the authors used Amazon Mechanical Turk (AMT) to come up with actions, write scripts, record videos and annotate the recorded videos. In doing so, they released a challenging third person dataset which includes a variety of video quality in terms of camera (position) as well as little bias in regards to action co-occurrences.

The dataset was later extended in [119] to include videos shot from an egocentric perspective allowing for the same actions to be viewed from both the third-person and first person perspectives. The relationship between these videos were explored in [118], through the use of an embedding space which allows for transfer of knowledge between the two recording modalities allowing for zero-shot recognition of actions in the opposite modality.

Actors, Objects and More Objects

The focus of object modelling within fine-grained action recognition can be seen as an extension of egocentric works given the fine(r)-grained nature of actions in egocentric datasets in comparison to third-person datasets. These approaches go further than simply detection of objects and explicitly model their temporal relationships: *I.e.* how are they interacted with/moved/changed? By doing this, models are able to discriminate between actions which could look similar when the active object is the same.

Baradel *et al.* [9] proposed a model which makes spatial-temporal reasoning of object interactions in a pairwise manner. They do this with the use of detecting objects, along with object masks, for videos. This allows for a function to be learned which represents the interaction of objects across differing frames. An RNN, further models *"long range reasoning"* between frames in a video from the pairwise representations. An overview of their approach can be seen in Figure 2.8 (from [9]).

They test their method on three different datasets including EPIC-Kitchens and Something-Something. The importance of the object relational reasoning can be seen in the ablation study, with its removal seeing a drop in performance across all three datasets. The RNN is also a beneficial addition leading to a smaller increase in accuracy.

Sun *et al.* [127], similarly focus on object interactions for the challenging task of fine-grained action recognition. They also model spatio-temporal relations between humans and scene elements. The proposed method, titled the actor-centric relation network, generates actor proposals in addition to extracting global features. The combination of global object features and actor proposals are used to learn relations between actors and

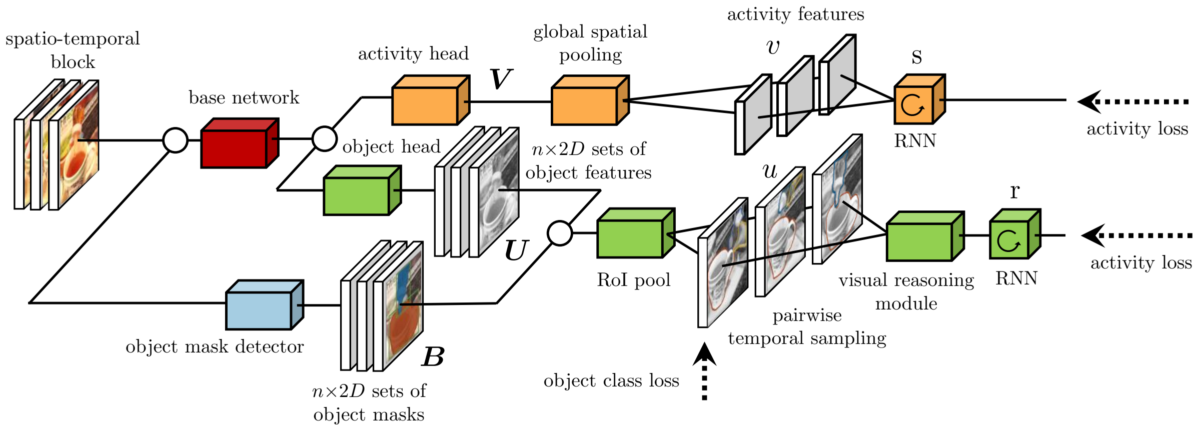


Figure 2.8: *Overview of the object relational network. Object features and object masks are extracted from frames which are paired together in the visual reasoning module to learn object relations over time using the notion of the arrow of time [100, 138]. These features are further augmented with activity features which learn from global and local motion. Figure from [9].*

objects[13] in a pairwise manner, similar to [9].

An in-depth ablation supports their design choices, namely the relational reasoning modules, but also the impact of temporal context when extracting features and the comparison of feature scaling. In the case of temporal context, *i.e.* number of frames used as input to the feature extractor, extracting global features doesn't necessarily translate to higher performance: They note the importance of correct reasoning of the relationships between actors and objects. Perhaps unsuprisingly for the short clip videos, increasing the temporal context leads to a direct increase in performance for their method. However, they find that earlier features generally give better performance, though a combination can achieve best performance.

Sudhakaran *et al.* [126] explicitly learn spatio-temporal discrimination of small objects, proposing Long Short-Term Attention which, as a method, extends LSTM blocks. The core part of their method is a pooling operation which learns to select attention maps from a learnt set depending on the input features. This attention is applied in two parts of the modified LSTM, on top of the visual features, using an RNN, and onto the output gate. The proposed Long Short-Term Attention modules are placed on top of spatial and temporal streams which are fused as in [33].

[13] The authors split the frames into an $n \times n$ grid of cells, extracting features for each. These are used as a simplified representation of object proposals within the scene.

The visual feature attention provides the largest boost in performance in their ablation study, but the output attention also gives a large increase. Using two-streams allows for each stream to learn complementary information, further increasing performance over an RGB long short-term attention model. Similarly to [33], learned fusion gives a slight boost in performance over the naïve method of late fusion. For EPIC-Kitchens, their model performs well on verb prediction which they ascribe to their model's ability to reason for longer periods of time over the baseline TSN [136] approach.

As seen in the egocentric action recognition background, object modelling remains a valuable tool in recognising fine-grained actions. However, methods are sure to model the temporal relationships of objects and how they are manipulated, as opposed to detection or simply extracting features over object trajectories.

Coarse-Grain Modelling for Fine-Grained Tasks

The other main approach towards fine-grained action recognition leads towards longer temporal modelling allowing for aspects of the activity as a whole to be encoded.

For example, Wu *et al.* [142] focus on learning from longer sequences around the action being recognised. Instead of only using frames from the video clip, their method also extracts long-term features from frames both before and after the clip (throughout the entire video). A feature bank operator then applies attention to the long-term features using the short-term features which are passed into the classification network for the current task.

They perform experiments on Atomic Visual Actions (AVA) [46], EPIC-Kitchens and Charades, noting the benefit of their approach on all three datasets. Interestingly, for AVA and Charades, the addition of a learned fully connected layer in the feature bank operator gives best performance, whereas for EPIC-Kitchens this is eschewed in favour of a simpler max pooling operation, which the authors believe is due to EPIC-Kitchens not having the human-human interactions present in the other two datasets. Also of note, they find that their performance increase on Charades is less than the other datasets as they are predicting on the comparatively coarser-grained, video-level labels.

Feichtenhofer *et al.* [34] also avoid modelling the actor-object relationships, instead proposing a two-stream network (SlowFast) which includes streams with different frame-rates, helping model both long and short-term dependencies within an action whilst also allowing for a lightweight stream. Both streams are fused via the use of lateral connections [32], with a unidirectional connection from the high-frame rate stream into the low frame-rate stream. A key benefit to their approach is the recognition accuracy the SlowFast network is able to achieve with a decreased cost in FLOPs, leading to state-of-the-art performance on four different datasets including Kinetics-[400/600] and Charades.

Concluding Remarks

As a relatively new task, fine-grained action recognition has seen two primary approaches to discriminate between similar actions: Object relational modelling and temporal modelling. Both approaches can be viewed as an extensions of ideas from egocentric action recognition which contained finer-grained actions in comparison to third-person action recognition datasets.

Nevertheless, fine-grained action recognition has remained a classification task. Even when open-vocabularies are present within datasets, works convert the open vocabulary labels into closed vocabulary labels via the use of a hierarchy (as in Something-Something [44]) or explicitly or clustered into close-vocabulary classes (as in EPIC-Kitchens [23]).

2.2.4 Action Recognition Conclusion

Beginning with a dataset containing only 6 different actions, [112], video action recognition as a task has undergone large additions in size, domain (both first person and third person) and coarse-ness of its actions. Early approaches used spatio-temporal interest points to generate local features which were encoded and classified using an SVM. For third-person action recognition, the rise of deep learning's popularity saw a renewed focus on spatio-temporal modelling whereas for third-person action recognition objects remained a key component.

Recently, fine-grained action recognition has become a popular and demanding task as the scale of datasets continued to increase to match deep learning techniques. Including a variety of similar tasks, fine-grained action recognition has, initially, seen approaches focus on relational object modelling (including using actors in the scene for third-person video) as well as longer temporal modelling.

Regardless, many works still focus on using a closed vocabulary of verbs (for coarse-grained) or verb-noun pairs as labels to perform action recognition. This thesis will explore how an increasing vocabulary size, including using an open vocabulary, changes the problem of action recognition and considerations that have to be made. Whilst some works within action recognition have used a natural vocabulary or even treated it as a multi-label problem all of these works still have the goal of using a closed vocabulary.

2.3 Information Retrieval

Information retrieval is a common task that can be seen in many different areas such as web search engines. As a generalisation, the aim of information retrieval is: Given a query item, rank (and select) all relevant items within a dataset.

In computer vision, this has classically been seen as an image-to-image retrieval task, one use-case is for matching buildings. For example, if a user has taken a picture of a building the aim would be to return other pictures of that building (these can include different viewpoints or during different seasons/change in weather conditions). In this case, the relevancy criteria is clear, a pair of images are deemed relevant to one another if their main focus is of the same building otherwise, the images are considered irrelevant.

With the advent of Google and YouTube cross-modal search, between vision and language, has become more common. Typically, websites see users input a search query in the form of a small amount of text and wish to retrieve visual example which is relevant. In this case the issue of relevance is a lot more nuanced. Methods to solve this problem need to take into account all aspects of the query sentence including presence of different words such as verbs, nouns, adjectives or other parts of speech which refine the search criteria.

Because of this, the cross-modal retrieval task is inherently an open vocabulary problem, where classes (as in action recognition) are not available. This section first discusses relevant works within the image retrieval literature, both within-modal and cross-modal, in section 2.3.1 before presenting cross-modal video retrieval works in section 2.3.2 and concluding notable findings from both tasks in section 2.3.3.

2.3.1 Retrieval in Images

Image retrieval has been explored as both a within-modal (image-to-image) and a cross-modal (text-to-image or vice versa) task. In this section, relevant works for both of these tasks will be presented as well as approaches which tackle the zero-shot retrieval task.

Finding Objects with Objects

Beginning with object retrieval, [92, 96], larger scale datasets included landmarks of buildings [8, 55, 98, 99] in addition to images, taken from Flickr [2], being used to construct datasets [51, 146]. Works are primarily focussed on instance retrieval which requires relevant items to be the same object instance. In early datasets, such as [92], only one of each object instance was present, but as the size of datasets grew it is common for multiple instances to be found within a dataset.

Radenović *et al.* [102] train siamese convolutional neural networks to perform image-to-image retrieval. Their method uses off-the-shelf networks which they train with both hard negatives (*i.e.* examples which look visually similar but are irrelevant) and hard positives (*i.e.* examples which look visually different but are relevant). They do this by replacing the final fully connected layers of a CNN and instead use maximum activations of convolutions (MAC) [43, 131] to find the embedded image representations (after the convolutional layers, different forms of max pooling are applied over regions of the image to get a global image descriptor). A contrastive loss is used to train the network:

$$L(x_i, x_j) = \frac{1}{2}\big(Y d(x_i, x_j) + (1 - Y)(\max(0, m - d(x_i - x_j)))\big) \tag{2.5}$$

where x_i, and X_j represent two images, m a margin, $d(a, b)$ a function which returns the distance between a and b, and Y the relevancy which is set to 1 if x_i and x_j are relevant and 0 otherwise. Intuitively, the loss attempts to pull relevant items close together in space whilst pushing irrelevant items to be at least as far as the chosen value of the margin parameter.

Their experiments show the importance of choosing pairs of images for their task, where, for the positive examples, the choice of relevancy can lead to large changes in performance. Firstly, when sampling hard negatives, they find that by ensuring that a high variability of hard negatives over simply choosing the hardest negatives gives higher mAP scores. Secondly, they construct the set of positive images via structure from motion (SfM) information and compare to selecting images which have a low MAC distance. Their results show that by allowing different viewpoints to create the set of positive examples higher performance can be achieved.

Diverging from previous approaches that used the notion of instance retrieval. Gordo and Larlus [42] instead tackle the semantic retrieval task whereby they wish to retrieve images depicting similar objects, not necessarily the same instances. They do this by using textual information in the form of image captions to learn a semantic-visual space with semantic similarity (heavily) correlating with visual/object similarity.

Similar to [135, 137], they propose using a triplet loss with multiple terms representing the different pairs of modalities (*e.g.* visual-text, visual-visual *etc.*, however, as they focus on the image-to-image retrieval task, which they forgo using the text-to-text loss). Their method, similar to their previous work in [43], uses an end-to-end triplet network with a ResNET [48] backbone upon which MAC is applied. Perhaps surprisingly, the addition of textual information during training on its own did not help the retrieval of images — only the addition of a caption during testing gave a noticeable increase in performance.

Trespassing the Boundary between Vision and Language

With the rise of social media, everyday images[14], represent popular uploads. Generally, on sites such as Flikr [2], images are often only labelled as tags or were provided with captions from the user[15], but some datasets, [49, 146], asked annotators to specifically provide captions which describe the makeup of the image.

Image-to-text retrieval became a natural task once techniques started looking into embedding cross-modal items close together [39, 40, 58, 64] for other tasks such as caption generation. The concept of semantic relevancy is hard to pin down for this task, as small changes in the caption can lead to them losing relevancy for an image. Because of this, most approaches attempt cross-modal image retrieval as an *instance retrieval* task, *i.e.* given an image return the corresponding caption, not any of the valid captions for other images.

[14] *I.e.* images which have been captured by amateurs on the fly, with little regard to framing *etc.*

[15] This is not beneficial as most images from social media sites have been captioned in a way that provides no information about the elements in the image. For example *"Us in Spain"* doesn't provide information about an image containing a couple standing on a beach.

Given the cost of annotating images with captions for a large-scale dataset, Gong *et al.* [40] instead propose to weakly learn a cross-modal embedding by using any available data, such as titles, tags, descriptions *etc.* In doing so, they propose a method which uses normalised canonical correlation analysis [39] and extended via an auxillary embedding of the learned features. Whilst they use CNN extracted visual features, the textual features are represented with the then-standard TF-IDF BOW representation.

Their main focus was using the above model to perform transfer learning from adding in a large amount of weakly-annotated images to a (smaller) dataset that had been fully annotated. Their experiments highlight two discoveries: Firstly, as the amount of fully supervised images to use as training data increases, the gap between (higher) performance of their proposed method and a fully-supervised method narrows. Secondly, the auxillary embedding used in their method, which increases the dimensionality of the embedding without degradation, is imperative for successfully using the combination of fully and weakly labelled images.

Performing on both the caption generation and image retrieval task, Kiros *et al.* [63] use an encoder-decoder network in the form of different LSTMs. To learn the embedding, they use an LSTM to embed the caption and a CNN combines different parts of the image to create the embedding. Interestingly, they use part of speech information, but only in the decoder part of the network for generating the caption, which is used with the embedded image information, not in the embedding.

Even when learning for the image captioning task, the latent embedding is found to perform well on the image retrieval task (both image-to-text and text-to-image retrieval). The retrieval results were also found to go against previous work in that both recurrent models they used were found to outperform models which used object detectors.

Wang *et al.* [137] tackle the image-to-text retrieval task and scale it with the addition of deep learning. They create two general cross-modal embedding networks which are able to perform different sub-tasks, such as phrase localisation (*i.e.* given a part of the caption, *"fire-pit"*, can a corresponding region in the image be localised?). They use features extracted from neural networks (Fast R-CNN features [38] for images and Word2Vec encoded with Fisher Vectors as in [64]), they construct two networks: An embedding network, which is trained with a triplet loss, and a similarity network, trained with a logistic loss. An overview of both methods can be seen in figure 2.9 (from [137]).

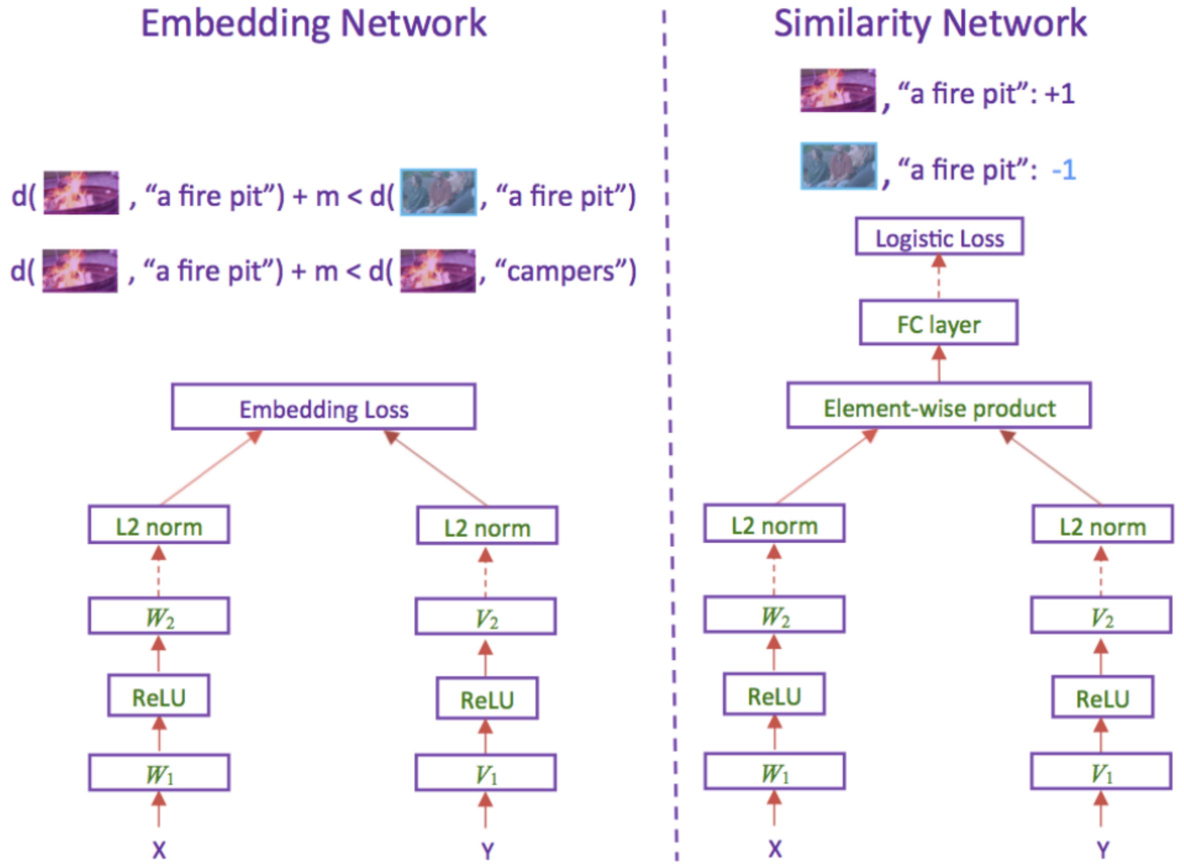


Figure 2.9: *Overview for the two models used for cross-modal retrieval between images and text in [137]. Left: The embedding network uses a triplet loss to ensure that positive examples are closer than negative examples (see equation 2.6). Right: The similarity network is created similarly, but replaces the triplet loss with the element-wise (Hadamard) product before a fully connected layer is used to predict similarity using a logistic regression loss. Figure from [137].*

The triplet loss is used to train the embedding network, presented below:

$$L(x, y^+, y^-) = max(0, m + d(x, y^+) - d(x, y^-)) \tag{2.6}$$

Intuitively, it calculates the bi-directional ranking between an image (x) and its relevant (y^+) and irrelevant (y^-) textual examples (in this case describing the image-to-text loss). The function for the text-to-image loss can be similarly constructed between a text caption and (ir)relevant images. Both triplet losses are used in the final loss function. In this work, additional triplet losses are added in which only items from a single modality are considered, *e.g.* for images:

$$L(x, x^+, x^-) = max(0, m + d(x, x^+) - d(x, x^-)) \tag{2.7}$$

Note: That x and x^+ represent different images/image patches in the formulation otherwise the loss is trivially 0. In their results, the similarity network is able to achieve comparable performance to the embedding network for the phrase localisation task, however, for all 'flavours' of the image-retrieval task (including within-modal retrieval) it achieves a very poor performance.

Retrieving the Unknown

Whilst retrieval can be thought of as intrinsically zero-shot (both the images and the captions at test time have not been seen before), this can be extended further via search and retrieval of visual elements that were not present in the training set. A cross-modal embedding between text and images allows for prior knowledge of word semantics to guide performance on unseen examples.

The two works of Zhang and Saligrama [150, 151] focus on zero-shot learning of images via the use of attributes/histogram proportions to describe unseen classes. For example, a *"car"* might be seen as a mixture of the classes *"truck"* and *"boat"*. Initially, in [150], they create two embedding functions for the source and target domains in which the latter includes zero-shot examples. By mapping classes and images into the same space, they are able to generalise semantic similarity information to the unseen domain. In the follow-up work [151], the authors propose modelling the zero-shot task as a binary classification problem, training a classifier to predict whether the class labels in the target domain are equivalent to those in the source domain.

Zellers and Choi [149] present the idea of using verbs as attributes to describe actions in the zero-shot setting for action recognition in images. Their approach uses a combination of word embeddings, from GloVe, with different lingustic and visual cues about the action. For example, drink is an activity with *low motion* in a *solitary* setting *with an object* that *uses the head*. By training a model which learns to embed dictionary definitions or word embeddings they are able to perform text-to-attribute prediction and image-to-verb prediction. They do this with the use of bi-directional Gated Recurrent Units [19] (GRUs) and a final fully connected layer maps the output into a shared embedding space.

Concluding Remarks

Image retrieval has been pursued in two main forms: cross-modal and within-modal tasks. The challenge for both tasks has been how to determine whether two items are relevant, with most works focussing on a instance-based approach. Nevertheless, within-modal information has been shown to improve cross-modal tasks in addition to textual captions being used as a way of going beyond instance retrieval. It can be seen that a successful retrieval method will therefore be aware of both within-modal and cross-modal relationships.

2.3.2 Retrieval in Videos

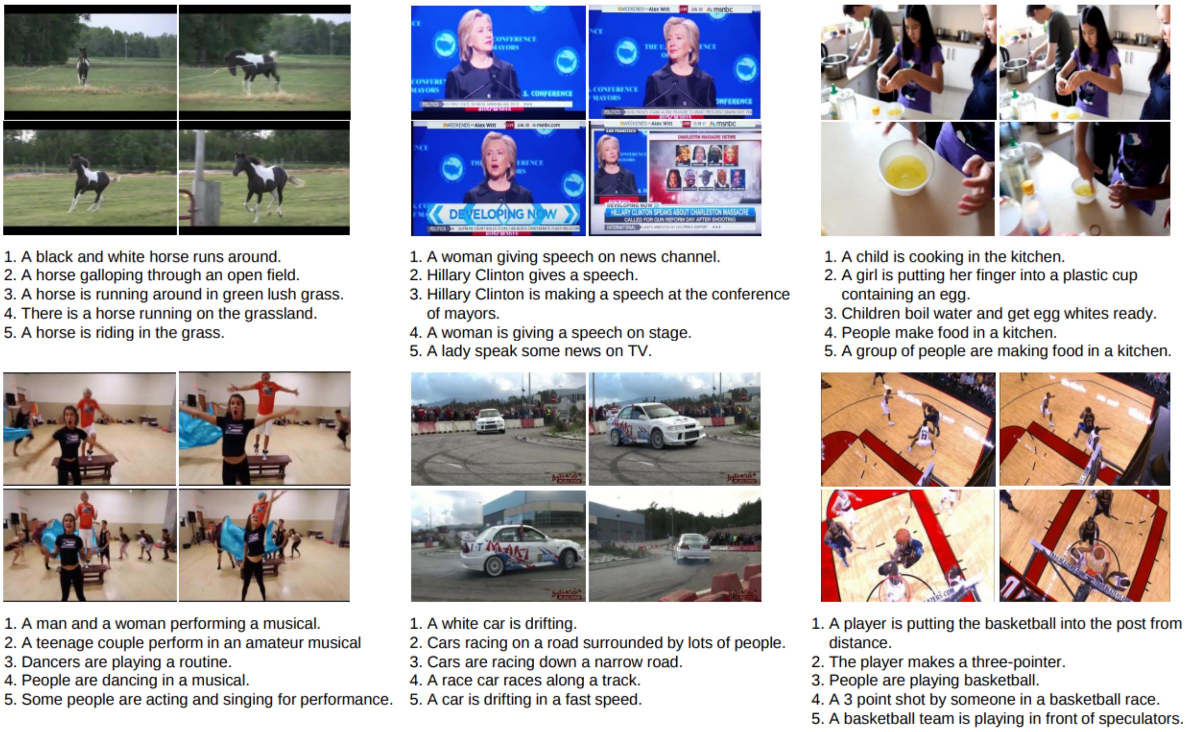
Video Retrieval is akin to image-retrieval in that queries from one modality to another are performed. Although, due to the inclusion of the temporal dimension in video, and the higher complexity in their representations, approaches tend to employ features instead of training CNNs end-to-end: These are either frame based and embedded using a recurrent network or, instead, extracted using a spatio-temporal network to ensure temporal information can be encoded. Additionally, video retrieval approaches can also be broken down into those which focus on the visual embedding or the textual embedding, in which the differences will be discussed during this section.

Similar to Image Retrieval, the notion of relevance can be difficult to define between videos and textual captions. As a result, approaches perform instance retrieval where relevant captions for a video are only those which were collected for that video, regardless of how similar two videos/captions might be. Within this thesis, the term *general video retrieval task* will be used to describe video retrieval in an instance-retrieval setting and an alternative to this will be presented in chapter 5.

This section introduces works which focus on general video retrieval in addition to presenting MSR-VTT, a video captioning dataset commonly used for this task, and HowTo100M, a recent large-scale, weakly labelled dataset.

MSR-VTT [144] is a video captioning dataset primarily designed for video-to-text tasks. It consists of $10,000$ videos which were compiled via 257 different text queries from 20 different categories[16] and used YouTube to collect 118 videos per query. For each video, an automatic colour histogram approach was used to separate each video

[16] The categories are: Ad[vertisement]s, Animals, Animation, Beauty, Cooking, Doc[umentary], Education, Food, Gaming, How To, Kids, Movie, Music, News, People, Science, Sports, Travel, TV Shows, and Vehicles.



1. A black and white horse runs around.
2. A horse galloping through an open field.
3. A horse is running around in green lush grass.
4. There is a horse running on the grassland.
5. A horse is riding in the grass.

1. A woman giving speech on news channel.
2. Hillary Clinton gives a speech.
3. Hillary Clinton is making a speech at the conference of mayors.
4. A woman is giving a speech on stage.
5. A lady speak some news on TV.

1. A child is cooking in the kitchen.
2. A girl is putting her finger into a plastic cup containing an egg.
3. Children boil water and get egg whites ready.
4. People make food in a kitchen.
5. A group of people are making food in a kitchen.

1. A man and a woman performing a musical.
2. A teenage couple perform in an amateur musical
3. Dancers are playing a routine.
4. People are dancing in a musical.
5. Some people are acting and singing for performance.

1. A white car is drifting.
2. Cars racing on a road surrounded by lots of people.
3. Cars are racing down a narrow road.
4. A race car races along a track.
5. A car is drifting in a fast speed.

1. A player is putting the basketball into the post from distance.
2. The player makes a three-pointer.
3. People are playing basketball.
4. A 3 point shot by someone in a basketball race.
5. A basketball team is playing in front of speculators.

Figure 2.10: *Example Video-Caption Pairs from MSR-VTT. For each Video-Caption pair, 4 frames of the video are shown along with 5 of the 20 corresponding captions. Figure from [144].*

into different shots. 15 Amazon Mechanical Turk (AMT) [1] workers were then asked to combine different shots to form video clips. To get the final list of video clips, the list was curated such that only a maximum of 3 video clips appeared from each video before a random selection of 10,000 videos clips was used as the final dataset. Each clip resulted in being between 10-30 seconds long.

In order to collect captions, for each video clip, 20 AMT workers were asked to watch the clip and provide a caption. Short captions and/or captions with little description were removed, leaving 200,000 different captions assigned in a 20:1 ratio to each video. Example video-caption pairs can be seen in figure 2.10 (from [144]), note how in some cases, for example the horse video (top left) that the captions can be very similar, whereas for other videos, for example the basketball video (bottom right) that captions can have varying amounts of detail depending on the annotator's knowledge (*"three Pointer" vs. "playing basketball"*).

In the original paper, MSR-VTT was evaluated for the task of video captioning. That is, given a video can a human-readable sentence be generated that describes the video? However, works which focus on the task of general video retrieval have started to use the dataset due to its large size and difficulty.

Using Semantics for Video Retrieval

Many works for video retrieval tend to target their efforts upon effectively modelling language and thus, whether explicitly or not, align the video embedding to the textual features.

Xu *et al.* [145] build a three-part model which aims to jointly model videos and sentences. They use a compositional semantics language model which first breaks down the caption into *subject, verb, object* triplets (SVO). These are then combined in a hierarchical fashion through the use of RNNs to create the language representation. The initial video features are extracted per frame from an ImageNet model, before being combined using a temporal pyramid and a two-layer multi-layer perceptron (MLP).

Their joint model is evaluated on three different tasks — Subject, Verb, Object prediction as well as both video-to-text and text-to-video retrieval — using the Microsoft Video Description (MSVD) [18] dataset. The results show that that their method improves over the baseline Canonical Correlation Analysis model both quantitatively and qualitatively, giving sensible retrievals even if the verbs or objects are incorrect.

Mithun *et al.* [88] tackle the problem of separating the video into objects and actions using RGB frames and flow/audio respectively. Then, they learn a video-text embedding for each; using text features which have been created from an end-to-end word embedding network with a final gated recurrent unit. To combine the two output embeddings, and perform retrieval, the similarity scores between each of the visual embedding vectors with the textual embedding are calculated and then summed together. They note that: *"It may be desired to use a weighted sum when it is necessary in a task to put more emphasis on one of the facets of the video"* however, they do not test this in their work.

MSR-VTT, along with MSVD [18], is tested for the tasks of both video-to-text and text-to-video retrieval. Their method performs well against others which learn a single embedding but, due to the differences in test sets used, their results cannot be compared with those in [83].

Using Video Components for Video Retrieval

Conversely, a larger emphasis can be put on the video projection function as well as the different representations used as input for the video such as RGB, Flow, audio or others[17].

The approach proposed by Yu *et al.* [147] instead focuses on predicting visual concept words for each video, thus allowing them to operate on a number of different video-to-language tasks including video-to-text retrieval. The method uses LSTMs with attention to predict concept words consistently across video frames. Different models can then be added to the concept word prediction in an ad hoc way to train for the different tasks. Relevant to this thesis, the video retrieval model uses an LSTM to encode the videos and applies information from the word concept model onto the textual input in the form of attention. The visual and textual streams are then projected together using a Compact Bi-linear Pooling layer [36] and a maxout layer [41] using a max margin loss. Their approach achieved the highest retrieval performance on the 2016 Large Scale Movie Description Challenge Dataset (LSMDC) [107].

In their follow-up work, Yu *et al.* [148] note that *"most previous approaches tend to focus too much on sentence information and easily ignore visual cues"*. To combat this, they present the Joint Sequence Fusion Model which creates a matrix from the Hadamard product between each word in the caption and each frame in the video. This representation, which captures all pairwise correlations over time between the visual and textual streams, is then passed into a convolutional hierarchical decoder that attempts to find meaningful matches in the embedding in an iterative fashion, alternating between positive pairs and negative pairs.

Similar to their previous work, the model can be used for multiple different tasks such as video retrieval, multiple-choice test and fill-in-the-blank for the LSMDC 2017 challenge and they also report results on MSR-VTT. Their ablation study shows the importance of using audio and, more significantly, the combination of attention and the convolutional hierarchy in their model across all tasks and datasets.

Miech *et al.* [83] use a multitude of different representations of video — including appearance, flow, audio and faces — in a mixture of experts model to create a video-text embedding space. Their approach can be seen in Figure 2.11 (from [83]) which treats the extracted feature from each video representation as an expert of that representation of video. Note how each representation of video creates a separate embedding (*i.e.* an appearance-text embedding, a motion-text embedding *etc.*). These embeddings are

[17] Note, normally these would be described as different *video modalities*, but representation is used here to differentiate between the vision-language modalities.

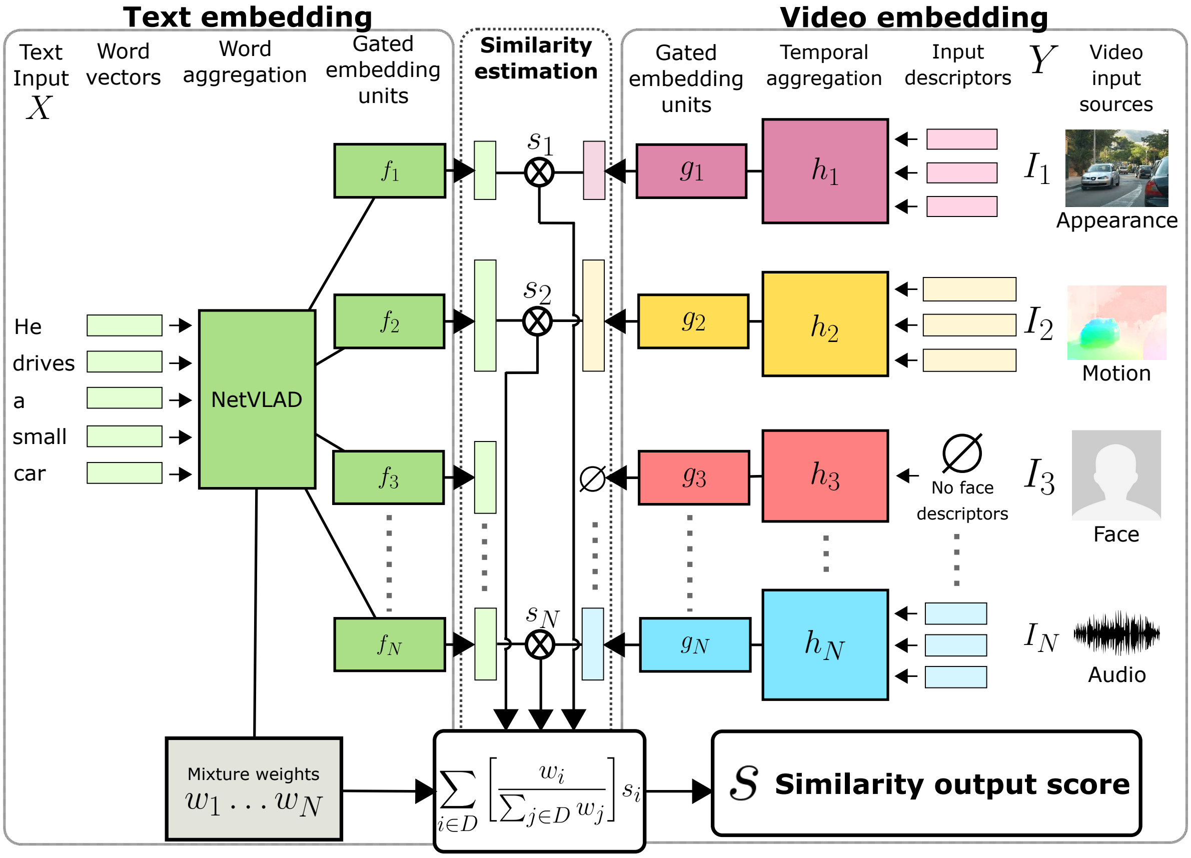


Figure 2.11: *Method diagram of the Mixture of Embedding Experts. Four different modality inputs can be used to embed a video along with a textual description. Figure from [83].*

combined by a set of mixture weights, which are predicted from the caption of the video. *I.e.* A caption such as *"A person is singing"* would potentially give a higher weight to the audio and face streams.

The method can also work in the absence of a specific representation (for example if the video had no sound or if the video doesn't include a person's face) by setting the mixture weights of the missing representation's embedding to 0 and re-normalising the weights of the other embeddings. In this way, the method has a large amount of flexibility and is able to scale to include more representations.

The mixture of embedding experts method was evaluated on three different datasets: MPII Movie dataset [107], MSR-VTT and Microsoft Common Objects in Context [74] (an image recognition dataset, also called MS-COCO). For the Large Scale Movie Description Challenge (LSMDC) of MPII, their method outperforms all previous methods, including the winners of previous years, when all modalities are used in addition to augmenting training with the MS-COCO dataset. From the other results, it can be seen that the augmentation provides large boosts to the proposed model and, where applicable or available, the addition of faces also gives a large boost to performance.

The Mixture of Experts idea is expanded in Liu *et al.*'s work [76] where they propose adding in more representations of video to be used as expert features[18] in addition to presenting a novel collaborative gating module, which replaces the learned mixture weights.

The collaborative gating module considers all pairwise interactions between representations (if available) to gate the relevant features using an attention-based mechanism. In the absence of a feature within the video, the same re-normalisation as above, in [83], is performed.

The large combination of representations proves useful, and the method outperforms all previous works on a variety of video retrieval datasets by large margins suggesting the importance of using the many representations of video to train with more information. Again, they find that the addition of face information provides a large boost to retrieval performance, with speech, OCR and audio also giving similar increases in performance over using appearance alone.

Learning from Weakly-Supervised and Noisy Data

HowTo100M [84] The HowTo100M dataset represents a large-scale video dataset with 136 million video clips which were collected from 1.22 million annotated instructional videos. The collection process was designed to be easily scalable and low effort that began via the acquisition of a list of activities from the WikiHow website[19]. They curate the tasks to create a final list of 23,611 visual tasks, removing those which include non-physical actions, such as *"feel"*, or abstract categories of instructions, such as relationships. Next, a search of YouTube videos was undertaken and popular videos with English subtitles are chosen. These subtitles are split into lines and become the captions of the video clips which are similarly split depending on when the caption occurred in the video. Because of this, automatic labelling, the dataset can be thought of as weakly-paired and, in general, are noisy (for example, only 51% of object/actions occur in a

[18] Including objects, scene info, actions, faces, optical character recognition (OCR), speech and audio.

[19] WikiHow allows users to upload instructional videos (as well as articles with accompanying images) of how to do various tasks.

clip).

Nevertheless, the collected dataset remains a useful resource given its size and diversity. The authors test a video-text embedding model, similar to [83], for a variety of different tasks and setups. Perhaps most notable is the increase in performance gained from training on the large dataset, and testing on MSR-VTT, with performance not saturating even when the full HowTo100M dataset is used. Additionally, when fine-tuning on the target datasets, the model proves to be even more beneficial for retrieval tasks.

Retrieving the Unknown in Videos

Zero-shot retrieval of videos is another possible task, in part due to the power of the learned cross-modal embeddings. Again, prior semantic knowledge from the unsupervised word embeddings plays a key role in ensuring that the learned embedding space is able to cope with unseen elements.

Directly embedding videos into a learned word embedding space was proposed in Hahn *et al.* [47], for the task of video retrieval, with the name Action2Vec. The word embedding is a pre-trained Word2Vec embedding which is directly compared to the output of a two layer hierarchical LSTM to better capture temporal information from the video.

By explicitly embedding videos into a Word2Vec embedding, the properties of such a space can be used, including vector arithmetic for analogy tests. In the paper they show how a video of *"playing piano"* minus the word embedding of *"piano"* and plus *"violin"* gives the video of *"playing violin"* as the nearest neighbour to the resulting vector. Due to coarser grained action datasets being used for testing (such as HMDB-51 and UCF101) the co-occurrences learned from the Word2Vec embedding represent meaningful similarities and the classes can be clustered successfully[20].

Dong *et al.* [26] propose a multi-level approach in which features from different levels of their pipeline are concatenated before a final layer projects them into the joint video-text space. Their hypothesis is that a powerful representation for video/captions, through the use of decomposition, is necessary for zero-shot retrieval. Figure 2.12 shows an overview of their method (from [26]). For each stream, visual and textual, three sets of features are extracted at different levels of the method. The first level corresponds to base features which are averaged temporally (image CNN features for the video stream and a one-hot

[20] Reasons as to why this doesn't hold true for fine-grained actions are explored further in section 4.3.

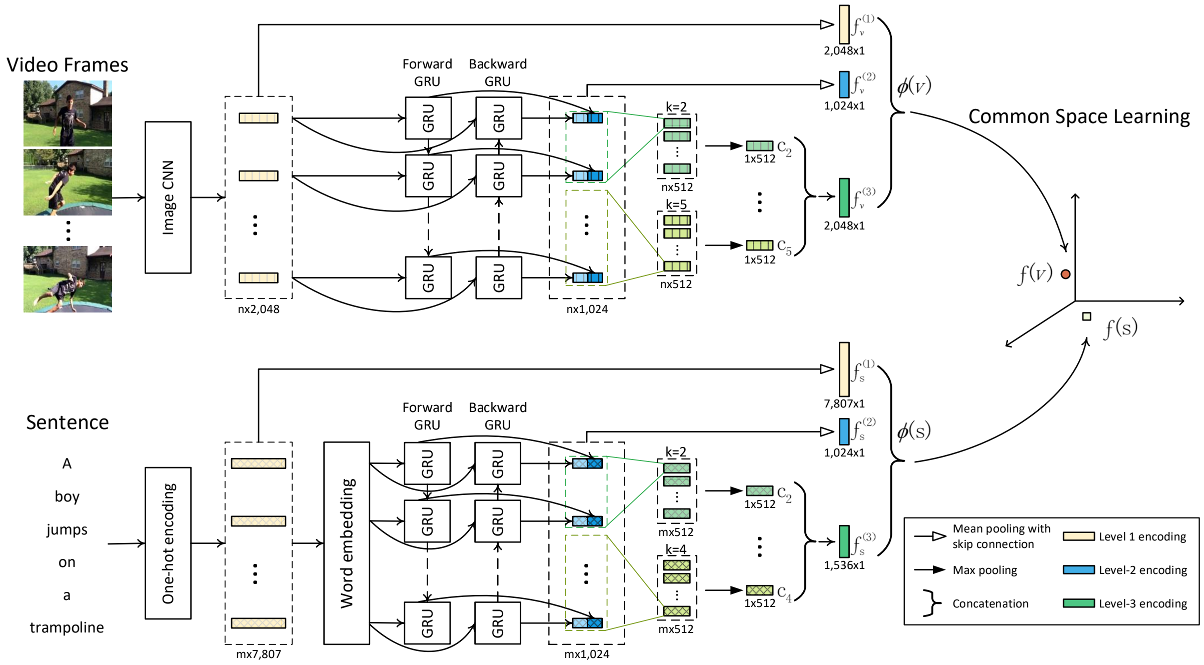


Figure 2.12: *Method overview for the Dual Dense Encoding method. Videos and text are embedded into a joint space using the concatenation of three different features extracted at different parts of the method. Figure from [26].*

encoding for text — similar to BoW). A bi-directional GRU is used to encode both past and future contextual information and the mean-pooled becomes the second layer encoding. Finally, the level 3 features are created via the GRU output undergoing a 1-d CNN transformation, using multiple filters of different scales, to find local patterns within the data. The resulting three levels of features are then concatenated before being projected into a cross-modal embedding space. Whilst not in the same vein as their work, the method proposed in chapter 5 of this thesis similarly uses video and caption decomposition to perform zero-shot retrieval.

Their experiments on MSR-VTT show the benefits of including all three encoding levels into the final representation. Interestingly, layer 1 (mean pooling) gives the largest sole performance on the video-to-text retrieval task, in comparison to level 3 (bi-directional GRU + 1D CNN) which performs best for the text-to-video retrieval task. Nevertheless, using all three layers gives retrieval scores that beat all previous works and baselines on MSR-VTT.

Concluding Remarks

Works focusing on the visual projection show the importance of including a large amount of data when training for the video retrieval tasks, whether through the addition of more video representations or simply more examples (even noisy ones). Even though it is widely accepted that CNNs can be information hungry, always requiring more and more data to learn successfully, with performance not saturating after 136 million videos using HowTo100M it shows how important the training set can be. Additionally, using multiple different representations of video has been consistently shown to allow for models to create a more meaningful projection into the embedding space.

2.3.3 Information Retrieval Conclusion

Information Retrieval can almost be thought of as a polar opposite problem when compared to action recognition: Retrieval is bi-directional whereas recognition is uni-directional, one uses an open vocabulary and natural language whereas the other (generally) uses a closed vocabulary. Importantly, the techniques and challenges within the information retrieval task represent useful knowledge when the vocabulary size is increased for action recognition.

A key issue with information retrieval approaches has been the question of: How to label two items as being (semantically) relevant? Most works forgo this question entirely, assuming that for each visual item there is only a single relevant caption. Classes, as defined in action recognition, could be used to provide a key benefit here in that, intrinsically, videos are considered semantically relevant within each class. With labels coming from an open vocabulary, where labels for videos in the same class are different, videos from the same class can be considered relevant if they belong to the same class, and similarly for captions. This notion of similarity will be discussed further in chapter 5.

2.4 Conclusion

This chapter has presented an overview of the relevant works this thesis builds on. As the work presented in this thesis relies upon Natural Language Processing and Lingustic knowledge, related work and terms were first introduced.

The first two methods are evaluated on the task of action recognition, and so next a background overview was given of this area, including the sub-tasks of third-person, first-person and, much more recent in its inception, fine-grained action recognition.

Finally, works that evaluate on the task of information retrieval were presented as chapters 4 and 5 evaluate on the task of action retrieval.

Chapter 3

Semantic Visual Embedding using a Graph

This chapter will begin to explore the complex space of verbs for action recognition. Specifically, as previous approaches used a small, closed vocabulary of verbs what are the challenges in breaking this tradition?

By extending the size of the vocabulary, classes become ambiguous. In previous works, when using a closed vocabulary setting for labelling a dataset, verbs were chosen to ensure that during the annotation process annotators wouldn't be confused between them. *I.e.* there is a single correct answer to label each video. This is no longer true when the vocabulary increases, which, considering the wealth of verbs that can be used to describe similar actions, can cause significant overlaps between classes.

Action recognition approaches currently treat the task as a classification problem using a one-vs-all approach. With significant overlaps between classes when using an open vocabulary this chapter shows that these standard approaches, such as Support Vector Machines (SVM), struggle as the vocabulary size increases.

The concept of an open vocabulary is explored first by collecting open vocabulary annotations for the Bristol Egocentric Object Interactions Dataset [22] (BEOID). Next, a method is presented which attempts to link actions both semantically and visually via the use of an underlying graph structure to deal with the ambiguity of the collected annotations. Results of this method are presented against two baselines: k-Nearest Neighbours and Support Vector Machines. Furthermore, the English lexical database, WordNet (see Section 2.1.1 for more info), is used to help relate the actions[1].

In detail: Section 3.1 discusses the classical approach, that video action datasets are labelled with using a closed vocabulary. It comments on the implications of expanding the number of labels under this regime and what happens when this is expanded. Section 3.2 includes information about the BEOID dataset for which annotations are collected with an open vocabulary. Next, section 3.3 contains the method in which the annotations were collected for BEOID in addition to their statistics. The method is presented in section 3.4 and experiments conducted in section 3.5.

3.1 Closed Vocabulary of Action Classes

Action recognition datasets have been classically collected using a closed vocabulary. Annotators were given a fixed set of verbs and nouns from which to choose class labels, creating non-ambiguous and non-overlapping classes to perform action recognition. However even when a pre-selected vocabulary isn't used, either from audio [5] or text [22], a single label is chosen by way of majority vote. In this way, uncommon annotations are treated as outliers or incorrect when they likely represent valid labels.

There are two main reasons for expanding the size of the vocabulary used for describing actions. Firstly, humans do not communicate with a fixed, non-overlapping vocabulary. Any task which requires human-computer interaction for action recognition would therefore have to include some mapping between the human's open vocabulary and the computer's closed vocabulary. Secondly, by including an expanded list of words, models can learn more about the action taking place. For example, when is an *"open door"* action a *"pull"* action and when is it a *"push"* action? For certain use cases, such as robotics, having a model which can understand the nuances between how actions can be performed is vital.

By expanding the vocabulary size, words which contain semantic overlaps will be collected (*i.e.* words with the same or similar meaning). These overlaps can cause issues for the standard task of action recognition due to the assumption that a single class is correct. Figure 3.1 shows this in more detail: *"Open"* can describe both the act of pulling

[1] Note: this work was accomplished in late 2015 prior to utilising deep learning solutions. The latter will be used in subsequent chapters

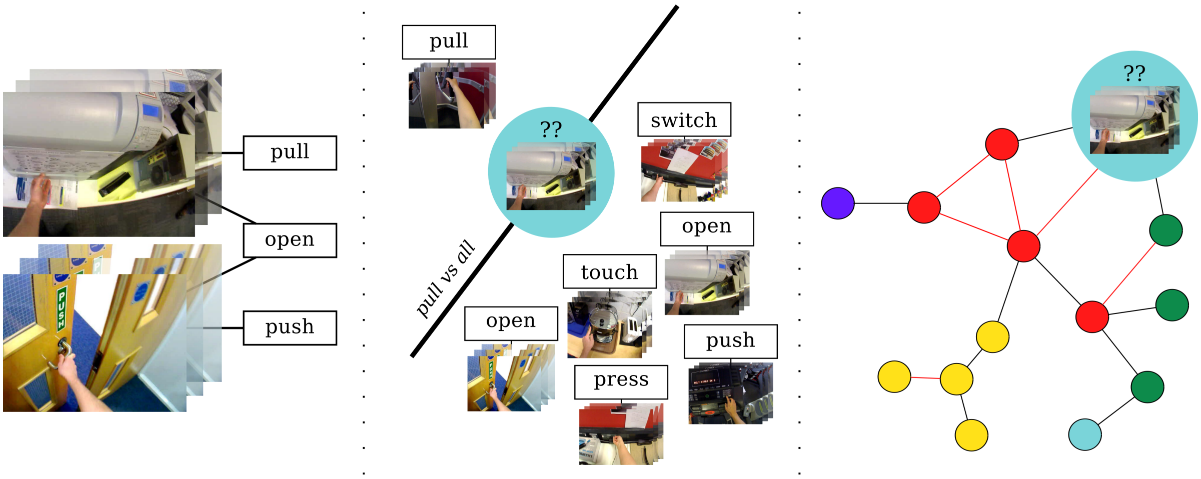


Figure 3.1: *Overview of the approach presented in this chapter. Left: When an expanded vocabulary is used multiple correct verbs can be used to describe an action.* E.g. *"pull drawer" can also be described by "open drawer". Middle: The classical method which uses one-vs-all classification can therefore struggle to give the correct answer. Right: In this chapter an embedding approach is presented in which the probability distribution for multiple classes can be found for an unknown video. Colours represent classes.*

open a drawer and pushing open a door. In the classification setup one-vs-all classifiers attempt to choose the correct class which breaks down when multiple classes can be considered correct. Note how the verbs shown in figure 3.1 describe object interactions. This thesis focuses purely on object interactions for two reasons: Firstly, they require an interaction to be performed and thus require temporal reasoning towards completion of a goal. Secondly, object interactions require a noun to describe the action (compared to say *"walking"*) and allow for an exploration into how similar/different verbs effect similar/different objects.

In this chapter open vocabulary annotations will be collected, expanding the vocabulary from what is used for standard video datasets and an embedding approach will be presented in which the probability distribution over potential labels can be returned.

3.2 Bristol Egocentric Object Interactions Dataset

The Bristol Egocentric Object Interactions Dataset (BEOID) [22] was collected in six different locations: kitchen, workspace, printer, door, cardiac gym and weight lifting machine. For each of the sequences within an area, the same script was used (*i.e.*

Area	Script Description	No. Videos	Objects
Cardiac Gym	Use the treadmill machine and the cycling machine.	9	treadmill, bicycle machine
Door	Use a keycard to go through a door.	10	Door handle, Door lock
Kitchen	Prepare a hot drink using a coffee machine.	10	tap, coffee machine, cup, sugar jar, spoon
Printer	Check paper is loaded in the printer.	10	printer drawer, printer keypad
Weight Lifting Machine	Use the weight lifting machine.	9	seat adjuster, pad adjuster, weight adjuster, weight lifting machine
Workspace	Plug in a screwdriver and interacting with objects.	10	Plug Socket, Box, screwdriver, charger, tape roll

Table 3.1: *Information of the BEOID dataset. The dataset was recorded in 5 different areas with the same script used within each area. The table includes the number of videos per area and the objects that were interacted with. Table modified from [22].*

all kitchen videos had the same object interactions and order of object interactions). The Kitchen, workspace, printer and door sequences had five different participants each recording a video twice (for a total of 10 different videos for each area) and the cardiac gym and weight lifting machine sequences were recorded by three different participants each with three recordings (for a total of 9 different videos). Table 3.1 includes an overview of the dataset in addition to the different scripts used and objects interacted for each of the six locations.

Each participant was wearing a head mounted camera[2] giving an egocentric or first-person view of the scene. This allows for an unobstructed view of the action and the objects being interacted with. However, the egocentric domain introduces a lot of motion in comparison to a fixed camera view because of the participant's head motion.

The BEOID dataset was released with object interaction bounding boxes and action descriptions that were not temporally localised. These annotations had been used purely to discover the task relevant objects and the different modes of interaction that could be applied to those objects. In order to train a model for action recognition on this dataset temporally localised annotations would have to be collected and, as discussed previously, this was done with no limitations on the verbs/nouns that annotators can choose. Details on how the annotations were collected can be found in the next section.

[2] ASL Mobile Eye XG

3.3 Annotation of Action Videos using Synsets

This section details the annotation process which was used to annotate BEOID with temporal action-level annotations using an expanded vocabulary of words.

The lack of any temporally localised action annotations on the BEOID dataset allowed for annotator choice to be incorporated in both the vocabulary that they choose in addition to the start and end points that are given for each action. It was hypothesised that, in both cases, annotators would not agree 100% in either case leading to different words being chosen as well as different temporal extents for each action.

As BEOID was to be collected using an open vocabulary, the size of the set of verbs used would be much greater than in closed vocabulary datasets. Compared to CMU-MMAC [130] or GTEA Gaze+ [31] the collected annotations would contain a large number of overlaps between verbs because of this. In order for a method to be able to understand and successfully learn from this information, external semantic knowledge was proposed to be incorporated within the annotations.

This took the form of WordNet, a lexical English database (which was introduced in section 2.1.1, this section contains information relevant only to this chapter). WordNet contains a number of different semantic relationships which can be used to relate two verbs. Specifically, the relevant relationships in this scenario are synonymy (whether two verbs have the same meaning) and hyponymy (whether one verb has a more specific meaning than the other verb). Additionally, the same verb can be used in different contexts and have different meanings. For example, the word *"hold"* could be used in the context of someone holding down a button or carrying an object in their hands. WordNet defines synsets, sets of words each with the same meaning, to model this behaviour. For example, the synset *"hold.v.1"* has the definition of *"keep in a certain state, position, or activity"* whereas *"hold.v.2"* has the definition of *"have or hold in one's hand or grip"*.

In order to ensure that the annotations collected for BEOID could be related using the two relationships from WordNet, annotators were asked to choose synsets in addition to the verbs that describe the video segment. The annotators were given the following instructions before using the annotator:

> - *You need to label the beginnings and endings of every object interaction throughout the video.*

*- The **beginning** of an object interaction is when you **first** recognise that the user is initiating **motion** to interact with the object.*

*- The **ending** of an object interaction is when you **first** recognise that the interaction is complete.*

- We are interested in actions; this means that you should split every activity (such as preparing a sandwich) into the actions of which [it] is composed (for instance, cut bread, spread butter, etc.*). An action itself may be interpreted as being composed of other sub-actions. For example, cutting a slice of bread requires a knife: If the action of picking the knife up is visible prior to the cut, then you should split them as separate actions.*

*- **Examples***

Let's consider a couple of examples to clarify what is the action granularity we need. Look at the videos "goodExample.avi" and "badExample.avi". Both belong to a sequence of a man preparing a salad.

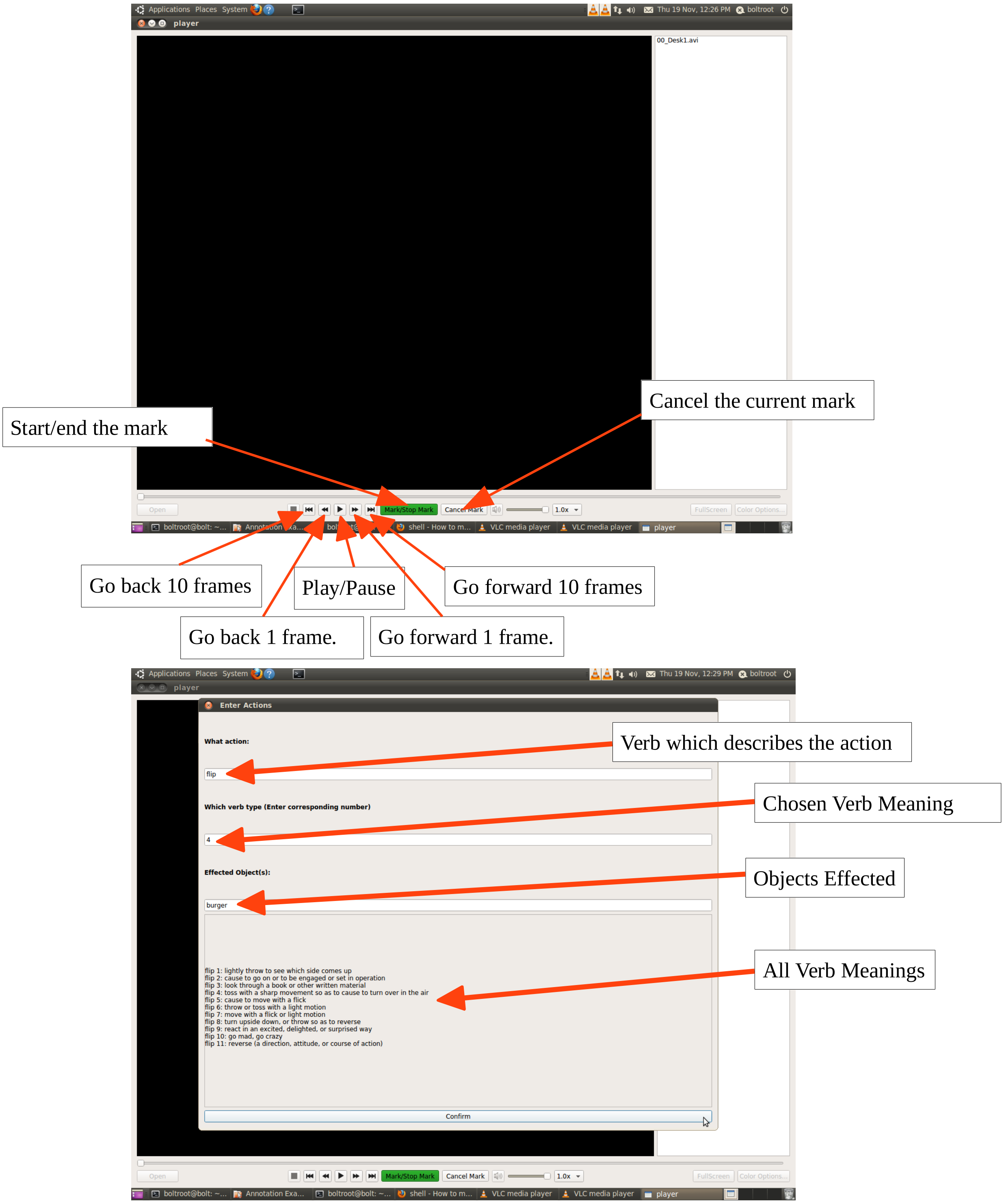
The first video shows the opening of a fridge, whilst the second shows both the opening of the same fridge and vegetables being picked up. The bad example is not a suitable segment for use because the two actions are clearly visible: that of opening the fridge and that of picking up the vegetables. Although you may think that opening the fridge is part of the picking action we need them separated since we are focusing on small granular actions.

Thanks!

This was accompanied by an instructions sheet that guided the annotator on how to use the annotation program which can be seen in Fig. 3.2. Note: The annotator would then choose a synset by entering its number or type a new verb. The annotator's chosen noun was also collected without synset information but wasn't used in this chapter.

Annotators were asked to annotate in sessions lasting up to half an hour at a time and were offered breaks in between annotating videos. Some annotators took part in multiple 30 minute sessions and, for each session, the annotators were paid £5.

During the annotation process, annotators had free reign to select any verb that described the action. Upon choosing a verb, a list of synsets were available to select from, whereby annotators were asked to choose the meaning that, in their own opinion, best matched



What action:

flip

Which verb type (Enter corresponding number)

4

Effected Object(s):

burger

flip 1: lightly throw to see which side comes up
flip 2: cause to go on or to be engaged or set in operation
flip 3: look through a book or other written material
flip 4: toss with a sharp movement so as to cause to turn over in the air
flip 5: cause to move with a flick
flip 6: throw or toss with a light motion
flip 7: move with a flick or light motion
flip 8: turn upside down, or throw so as to reverse
flip 9: react in an excited, delighted, or surprised way
flip 10: go mad, go crazy
flip 11: reverse (a direction, attitude, or course of action)

Figure 3.2: *The instructions given to BEOID annotators on how to use the annotating interface.* ***Top:*** *The normal view of the player had the video in view and allowed users to watch the video as well as skipping forward/backward throughout the video. Once a mark had been started and ended the action screen would appear.* ***Bottom:*** *The action screen allows the annotators to enter their chosen verb. A list of synsets and their definitions would then be given from WordNet.*

the action that had taken place. If no synsets existed for the chosen verb which matched the context of the video, then they were asked to provide a new verb with a synset that was correct. Note, this meant that the annotators were limited in some part on which verbs they could choose, but, due to the number of synsets in the verb hierarchy, annotators rarely had to change their verb choice.

In total, 21 different native English annotators were asked to annotate the dataset and contributed $1,225$ different action segments. During the process, 10 different non-native speakers also participated in the annotations, but it was found that whilst their choice of verb was often correct, their choice of synset was not. In the end, the non-native annotations were not used.

One annotator annotated every video to have a consistent set of labels across all videos. They were given detailed instructions to have a set of annotations that could represent a *"gold standard"*.

3.3.1 Collected Annotation Statistics

Each annotator annotated an average of 7.29 ± 11.74 videos[3]. Of the 58 different videos in BEOID the maximum number of times a video was labelled was 5 and the minimum was ensured to be 2 (with one being the annotator which annotated every video). On average, a video was labelled by 2.64 ± 0.74 annotators and each segment had an average length of $1.6s$[4]. The dataset could therefore be constructed using the standard vocabulary set-up of a single verb per class, or use the multiple verbs chosen per class that will be used in this chapter.

Figures 3.3 and 3.4 respectively show the average number of actions annotators chose per video and the number of actions annotators chosen per location. The gym sequences contained a high number of annotated actions, due to the repetitive nature of the tasks compared to making a cup of tea in the kitchen sequence.

As a whole, for the 58 different actions, the annotators chose 140 different verb synsets which consisted of 97 different verbs. 32 difference verbs were chosen with two or more

[3] This is somewhat skewed due to the single annotator that annotated every video. Not including this annotator the average becomes 4.75 ± 3.10.

[4] Comparatively, CMU-MMAC and GTEA Gaze+ had average action lengths of $8.7s$ and $2.0s$ respectively.

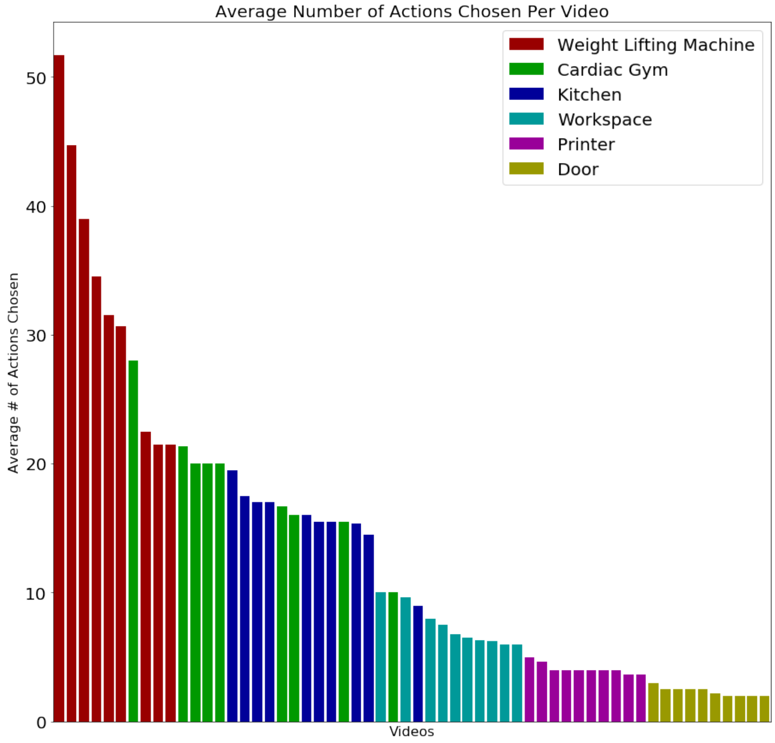


Figure 3.3: *Average number of actions chosen per video for BEOID.*

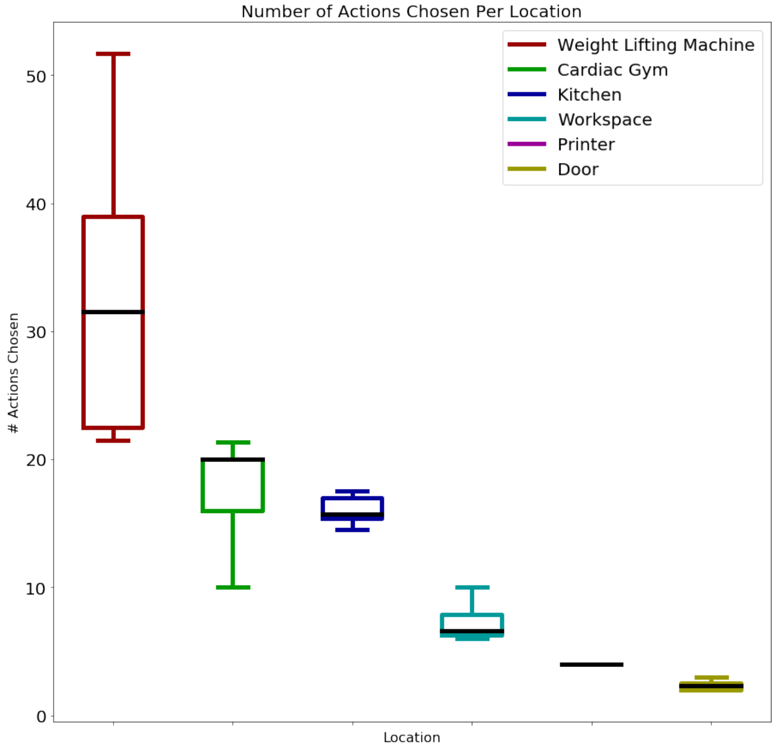


Figure 3.4: *Number of actions chosen by annotators per location.The two gym sequences, "Weight Lifting Machine" and "Cardiac Gym", contain the most number of actions per video. Whereas the door sequences are very short with only 2-3 actions per video.*

Rank	Synset	Count	Description
1	*"press.v.1"*	363	*"press (exert pressure or force to or upon)"*
2	*"push.v.1"*	242	*"move with force, "he pushed the table into a corner""*
3	*"pull.v.1"*	219	*"cause to move by pulling"*
4	*"pick up.v.1"*	67	*"take and lift upward"*
5	*"place.v.1"*	65	*"put into a certain place or abstract location"*
6	*"insert.v.1"*	63	*"put or introduce into something"*
7	*"pull.v.10"*	49	*"operate when rowing a boat"*
8	*"open.v.1"*	44	*"cause to open or become open"*
9	*"take.v.4"*	40	*"get into one's hands, take physically"*
10	*"relax.v.2"*	36	*"make less taut"*

Table 3.2: *The top 10 selected synsets in the collected BEOID annotations along with their definitions. Note the two "pull" synsets with one synset being used solely on the rowing machine.*

synsets. On average, 1.44 synsets were chosen per verb with a standard deviation of 0.84. Additionally, the annotators chose 140 different nouns. The most common synset, with 363 different instances, was *"press.v.1"* which has the meaning of *"exert pressure of force to or upon"*. This was most commonly paired with the noun of *"button"* (328 instances). The top 10 synsets that were chosen can be seen in Table 3.2 and the top 10 chosen verbs/nouns can be seen in Table 3.3. Note that both distributions are long-tailed with a small number of classes making up a large majority of the dataset (the top 10 verb synsets and verbs make up 60.9% and 66.6% of the dataset respectively).

Figure 3.5 shows the length of the actions in seconds that each annotator chose per location. Between locations the annotations are consistently $1 \sim 2$ seconds long, but the greatest amount of variance can be seen in the cardiac gym sequence. *"Run.v.1"* with the meaning of *"move fast by using one's feet, with one foot off the ground at any given time"* was the longest action annotated with 74 seconds, however, it was only annotated a single time by the annotators. On average each labelled action had a length of 2.00 seconds highlighting the fine-grained nature of the actions.

Additionally, Fig. 3.6 shows the lengths annotators chose for the top 10 synsets and verbs. In both cases, the most common synset/verb (*"press(.v.1)"*) has the lowest median length at just over 1 second. Interestingly, there is very little shift in median value between the synsets and their verbs suggesting that other meanings of the same verb have a similar length. For example, *"push.v.1"*'s distribution of lengths (top in Fig. 3.6) is very similar to that of *"push"* (bottom in Fig. 3.6 highlighting that all *"push"* actions have a similar length regardless of verb meaning and/or context.

Rank	Verb	Count
1	*"press"*	364
2	*"pull"*	302
3	*"push"*	250
4	*"pick up"*	86
5	*"insert"*	67
6	*"place"*	65
7	*"open"*	46
8	*"take"*	44
9	*"turn"*	38
10	*"relax"*	37

Rank	Noun	Count
1	*"button"*	413
2	*"rowing machine"*	279
3	*"handles"*	222
4	*"cup"*	81
5	*"jar"*	57
6	*"plug"*	56
7	*"spoon"*	51
8	*"tape"*	48
9	*"hand bars"*	41
10	*"mug"/ "tap"*	38

Table 3.3: ***Left:*** *The top 10 collected verbs (left) and nouns (right) from the open vocabulary annotations of the BEOID dataset. Both distributions exhibit features of a long tail distribution.*

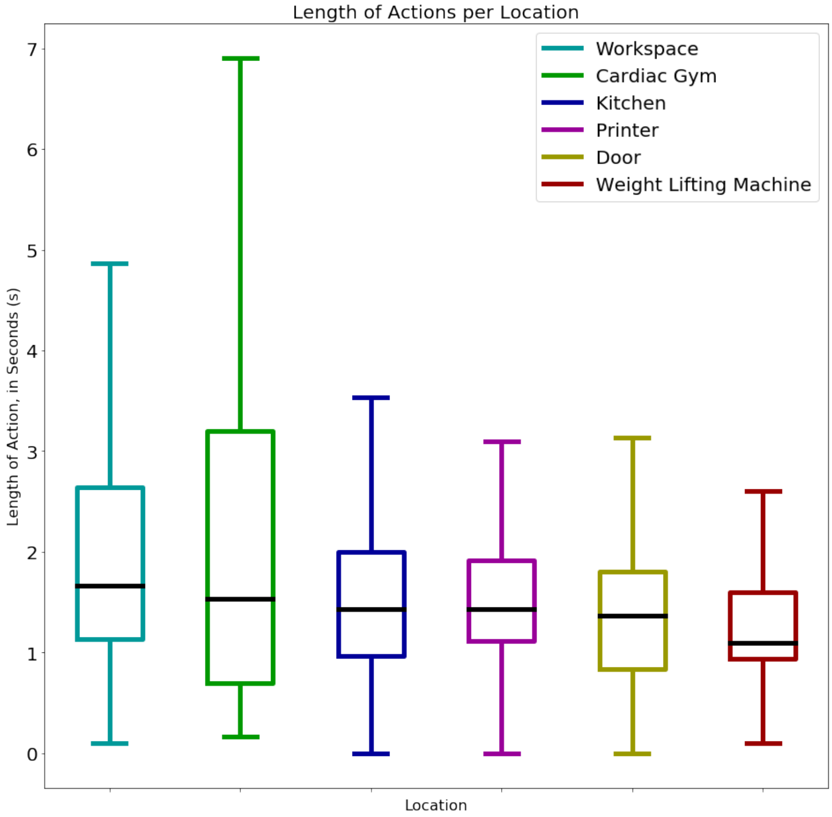
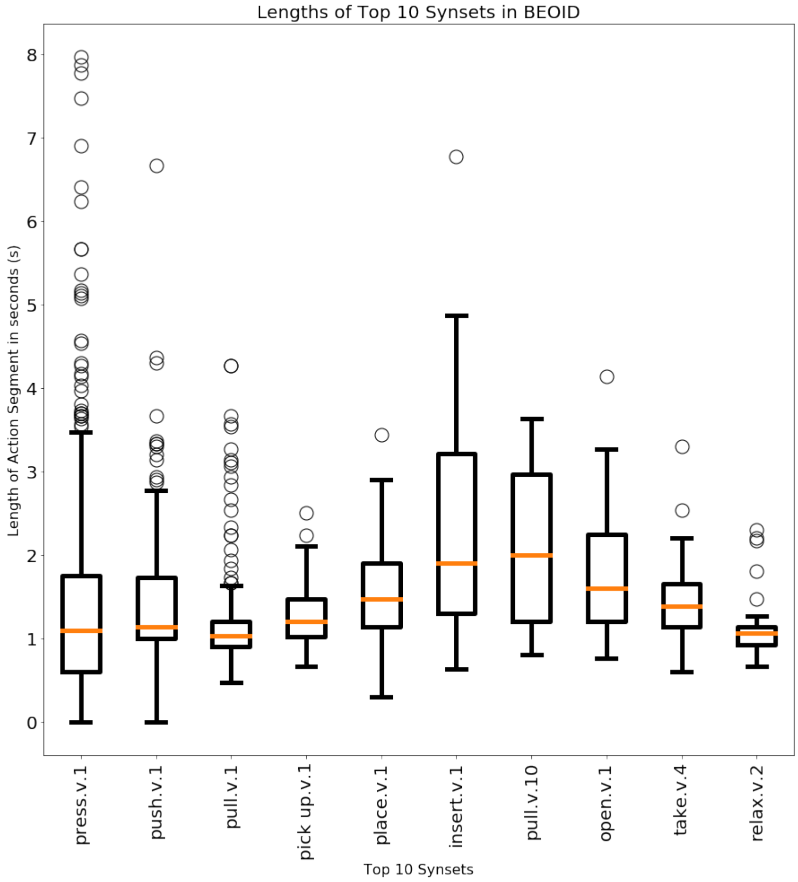


Figure 3.5: *The length of actions, in seconds, that annotators chose per location.*



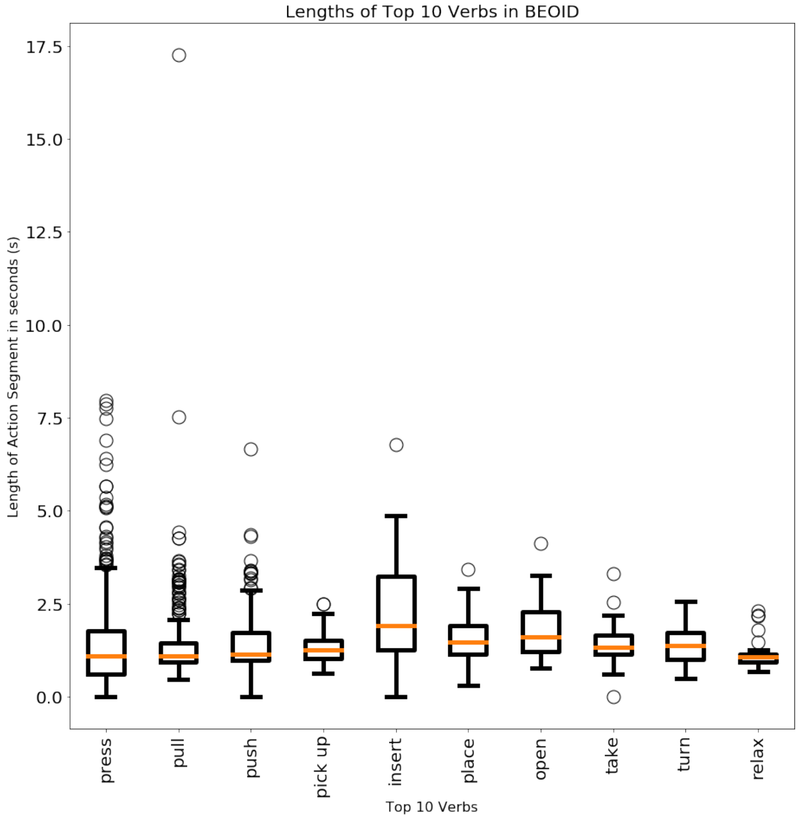


Figure 3.6: *The lengths annotators chose for each action of the top 10 synsets (top) and the top 10 verbs (bottom).*

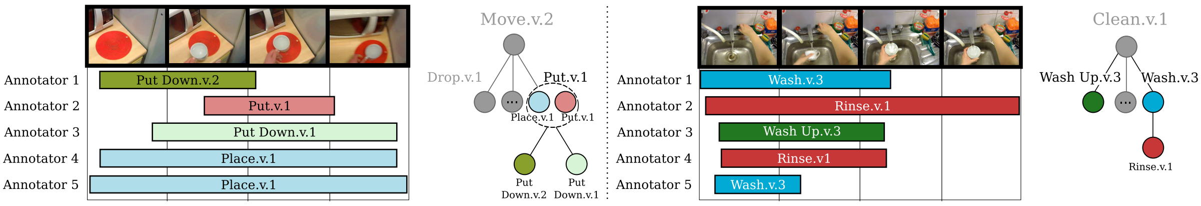


Figure 3.7: *Example annotations from five different annotators for two different actions from a kitchen sequence showing "put down cup" (left) and "wash cup" (right). The temporal bounds are also shown for each annotator as well as how the synsets are linked within the WordNet verb hierarchy via synonymy (e.g. "put.v.1" and "place.v.1") and hyponymy (e.g. "wash.v.3" and "rinse.v.1").*

Finally, example annotations from a kitchen sequence are provided in Fig. 3.7. In these two examples, the annotators have chosen many synsets for each action, all of which represent a correct label. It is also interesting to note that even when annotators choose the same synset (or even verb), the start and end points of the action don't necessarily align. A subset of the WordNet verb hierarchy is also included in the figure, showing that the verbs chosen by the annotators were linked via synonymy (*e.g.* *"put.v.1"* and *"place.v.1"*) as well as being linked via hyponymy (*e.g.* *"wash.v.3"* and *"rinse.v.1"*).

3.4 Semantic Visual Graph Embedding

In this section, a method will be presented that aims to use semantic information with the overlapping nature of the open annotations that were collected in the previous section. The method, designated as SEMBED, is an adaption of the method presented by Fang and Torresani [27] which was originally created for images using the noun hierarchy in WordNet.

Firstly, the method on how to create a semantic-visual embedding in the form of a semantic-visual graph is presented in section 3.4.1 before details on how to embed an unknown video into the network is given in section 3.4.2. Finally, information on how the synonymy and hyponymy relationships in WordNet can be used to assist in training the method in section 3.4.3.

3.4.1 Learning a Semantic Visual Graph

At a high level, the aim of the Semantic-Visual Graph (SVG) is to embed videos which have a high semantic similarity and/or a high visual similarity close together. The SVG does this in three ways: Firstly, videos which are semantically linked via WordNet will be linked in the SVG. Secondly, videos which are visually similar (*i.e.* the video's features are similar) will also be linked within the SVG. Finally, edges between videos are weighted so that the similarity between different pairs of videos can be found. This section first details the steps in creating an undirected graph, SVG$_u$, before normalising the edge weights to create the directed graph SVG which has the properties listed above.

Creating the Undirected Semantic Visual Graph (SVG$_u$)

Formally, given a set of videos, X, with x_i representing the ith video, each video has a corresponding class label C from the set of labels Y, a graph SVG$_u$= (V, E) will be created where V is the set of vertices and E the set of edges that link two vertices. As the SVG$_u$ is embedding videos, the set of videos X makes up the set of vertices V. An edge linking two videos x_i and x_j is thus defined as $e_{i,j}$. In an effort to shorten notation, $e_{i,j} \in$ SVG is used to show that the edge $e_{i,j}$ is a member of the set E belonging to the graph SVG.

In order to relate videos semantically, a binary function, $AX(x_i, x_j)$, is defined which will return *true* if x_i and x_j are semantically related and *false* otherwise. The details of AX are given in section 3.4.3 in which different variants are described for the different levels of semantic relevancy between two videos. Edges are created to link two semantically similar videos as follows:

$$e_{i,j} \in \text{SVG}_u \iff AX(x_i, x_j) = \textit{true} \tag{3.1}$$

The edge $e_{i,j}$ is given a weight $w_{i,j} = D_v(x_i, x_j)$ where $D_v(x_i, x_j)$ returns the distance between the visual descriptors of the two videos x_i and x_j[5].

[5] Euclidean or cosine *distance* were both evaluated and found to give similar results.

The next stage of building the SVG$_u$ is to add in the connections between videos which are visually similar. First, the function *rank* is defined such that it returns the rank of distance between two videos compared to all video pair-wise distances in the dataset as a whole:

$$rank(D_v(x_i, x_j)) = n \iff D_v(x_i, x_j) = min_n(D_v(x_k, x_l)) \qquad \forall x_k, x_l \in \text{SVG}_u \wedge AX(x_k, x_l) \neq true \tag{3.2}$$

where $min_n(D_v(x_k, x_l))$ returns the nth smallest visual distance between every node which isn't semantically related. The top m visually closest pairs are then linked within the SVG$_u$ as follows[6]:

$$e_{i,j} \in \text{SVG}_u \iff rank(D_v(x_i, x_j)) \leq m \tag{3.3}$$

Again, the edge $e_{i,j}$ is given the weight $w_{i,j} = D_v(x_i, x_j)$ and m is a hyperparameter which is chosen during training.

To further ensure that the graph is connected and as few unconnected cliques as possible exist, videos are also linked to their most visually similar, semantically distinct node. To do this, $rank_i$ is defined to give the rank of distances between the video x_i and its neighbours:

$$rank_i(D_v(x_i, x_j)) = n \iff D_v(x_i, x_j) = min_n(D_v(x_i, x_l)) \qquad \forall x_l \in \text{SVG}_u \wedge AX(x_i, x_l) \neq true \tag{3.4}$$

Finally, the nodes are linked to their most visually similar, semantically different pair:

[6] it is possible that two edges have the same weight which would lead to them sharing the rank in equation 3.2, however, this issue is resolved by choosing the top m connections in 3.3.

$$e_{i,j} \in \text{SVG}_u \iff rank_i(D_v(x_i, x_j)) = 1 \qquad \forall i \neq j \quad (3.5)$$

Note that this differs from [27] as the top m connections are found to be enough to allow the graph to be mostly connected for their purposes. This was due to the difference between the verb and noun hierarchies in WordNet, notably the smaller number of hyponymy connections in the verb hierarchy.

Creating the Directed Semantic Visual Graph (SVG)

The undirected SVG, SVG$_u$, can now be converted into a directed graph via normalisation of the weights for each node. At a high level, the aim for the directed Semantic Visual Graph, SVG, is that the sum of weights leaving a node add up to 1. Thus, each undirected edge is first broken down into two directed edges:

$$\forall e_{i,j} \in \text{SVG}_u \Rightarrow \{e'_{i,j}, e'_{j,i}\} \in \text{SVG} \quad (3.6)$$

where $e'_{i,j}$ is the *directed* edge that links node i to node j. Note that the nodes themselves, *i.e.* the set V, is unchanged between SVG and SVG$_u$. The weight $(w_{i,j})$ of the undirected edge $e_{i,j}$ is initially used as the weights for the two new edges, $e'_{i,j}$ and $e'_{j,i}$. Next, for each node x_i, the edge weights are normalised as follows:

$$w'_{i,j} = \frac{1/w_{i,j}}{\sum_k 1/w_{i,k}} \quad \forall e'_{i,k} \in \text{SVG} \quad (3.7)$$

where $w'_{i,j}$ is the weight of the directed edge $e'_{i,j}$. The reciprocal of the original weights are used so that the edge weightings represent the similarity between two videos (*i.e.* low distance is converted to a high similarity). By normalising the weights per node to add up to 1, and thus represent a probability distribution, the weight of each edge represents the probability of choosing it when performing a random walk through the graph. Additionally, a path with high probability will correspond to having small visual distances between videos that make up the path.

3.4.2 Embedding in an SVG

The previous section detailed how to create a semantic visual graph such that videos are linked both semantically and visually. The graph is directed and weighted so that each edge represents the similarity between two videos.

In order to learn the class of a new, unknown video, and thus embed it into the graph, the following steps are carried out: First, the z closest neighbours to the unknown video are found within the graph. Second, a Markov Walk is undertaken for t steps throughout the graph and a probability over all classes can be found. Finally, the class with the highest probability is chosen as the label for the unknown video.

Specifically, to embed an unknown video x_u, a set $\mathcal{R}$ is created so that it contains the z closest neighbours that are already embedded within the graph.

$$\mathcal{R} = \{x_i \in \text{SVG} \,|\, rank(D_v(x_u, x_i)) \leq z\} \tag{3.8}$$

The unknown video, x_u, is then embedded into the graph linked to the neighbours in $\mathcal{R}$. *I.e.* $x_u \in$ SVG and $\{e_{u,k}, e_{k,u}\} \in$ SVG, $\forall x_k \in \mathcal{R}$. The weights between the unknown video and the z closest neighbours are all normalised as in equation 3.7 (Note that this updates the weights of all edges connected to the nodes in $\mathcal{R}$).

Following this embedding, the class of x_u can be found via a Markov Walk that traverses the nodes in the graph to estimate the probability of the unknown video belonging to a certain class. As the weights of edges between nodes have been normalised, the probability to traverse an edge between two nodes is given simply by the weight of the edge between the two nodes. *I.e.* $P(x_j|x_i) = w'_{i,j}$, where $p(x_j|x_i)$ denotes the probability of reaching node x_j from x_i.

The probability distribution of reaching a certain node can be found given the Markovian assumption and a fixed number of steps, denoted by t using the following:

$$P(x_{i+t}|x_u) = \prod_{x_i \in \mathcal{R}} \big(P(x_i|x_u) \prod_{j=1}^{t} P(x_{i+j}|x_{i+j-1})\big) \tag{3.9}$$

Here, x_{i+j} denotes the node in the path j steps from the original node x_i. In order to efficiently calculate the probability distribution $P(x_{i+t}|x_u)$, the vector of probabilities q is constructed as below:

$$q(i) = \begin{cases} P(x_i|x_u), & \text{if } x_i \in \mathcal{R} \\ 0, & \text{otherwise} \end{cases} \tag{3.10}$$

The adjacency matrix of SVG, labelled as A, is given as follows: $A(i,j) = w'_{i,j} = p(x_j|x_i)$[7]. Therefore, equation 3.9 can thus be calculated using:

$$P(x_{i+t}|x_u) = q^T A^t \tag{3.11}$$

where q^T is the transpose of q and A^t is the matrix A raised to the tth power, representing the t steps of the Markov Walk.

Next, the probability of each class, C, can be found by summing over the different probabilities of each node. *I.e.* The probability of the video x_u belonging to a class is given by the sum of probabilities of reaching all videos in the graph within t steps that belong to that class. Specifically:

$$P(C|x_u) = \sum_{x_{i+t} \in C} P(x_{i+t}|x_u) \tag{3.12}$$

[7] Note that this matrix is asymmetrical due to the normalisation of the weights for each node.

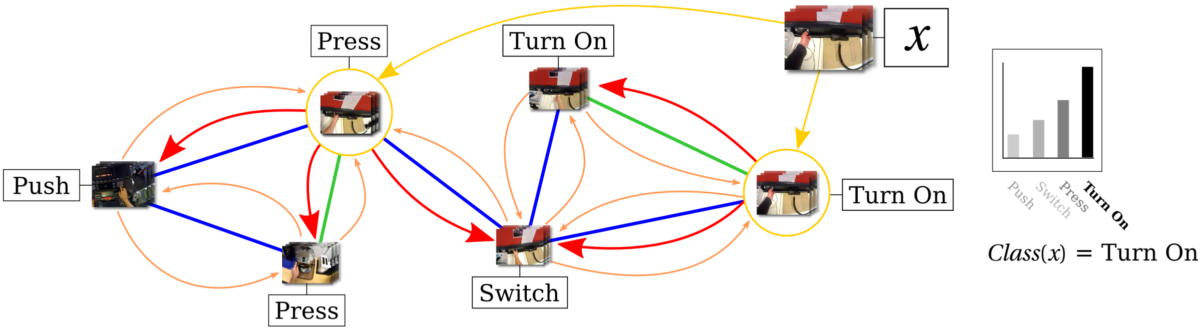


Figure 3.8: *Overview of the approach used to embed an unknown video into the trained SVG. Videos in the SVG are linked either semantically (in green) or visually (in blue) — In this case videos are linked semantically only if they share the same verb label. For an unknown video x, the two closest neighbours are found and then the probability distribution of ending at a certain node after two steps is found (first step in red, second in orange). The class probability distribution is shown to the right and a final class is chosen for the unknown video, in this case "turn on". Note that the graph doesn't show the directed edges in the graph for simplicity.*

where C represents a class. Note that the set of possible classes changes depending on the semantic relevancy function AX that is being used (see section 3.4.3 for more details).

The semantic label of x_u is chosen by selecting $\arg\max_C P(C|x_u)$. An overview of the entire embedding method can be seen in Fig. 3.8. The figure has the values of the hyperparameters z (the number of neighbours found for x_u) and t (the number of steps in the graph) to be both equal to 2.

3.4.3 Semantic Relevancy from WordNet

As discussed in section 3.3, the annotations were collected with synset information from WordNet. Using this, in addition to the different relationships in WordNet, the different verbs collected can be grouped together.

The two main relationships that are used in this chapter are *synonymy* and *hyponymy*.

- **Synonymy:** Two words are synonyms if they have the same exact meaning, *i.e.* they are completely interchangeable. In WordNet, synonymy is represented by multiple lemmas being contained within the same synset. For example, the synset *"put.v.1"* with the meaning *"put into a certain place or abstract location"* has the following lemmas, and therefore synonyms: *"put"*, *"set"*, *"place"*, *"pose"*, *"position"* and *"lay"*.

- **Hyponymy:** Two words are hyponyms if one has a more specific meaning of the other. In WordNet the hierarchy of words includes hyponymy relationships between them allowing for a tree structure with generic verbs as the root node and specific verbs as the leaf nodes. For example, the synset *"wash.v.3"* has the definition of *"cleans with a cleaning agent, such as soap, and water"*. *"Rinse.v.1"*, which is a hyponym of *"wash.v.3"*[8] has the meaning *"wash off soap or remaining dirt"*.

Using these two relationships different levels of semantic relevancy can be defined.

Firstly, none of the relationships can be used, the synsets can be treated as independent classes with different meanings. Of course, this is naïve given the structure of WordNet but it can be treated as a baseline used only for comparison. This level of relevancy is designated as *Action Meanings*, or AM.

Additionally, another relevancy can be created without using any of the relationships from WordNet: Not using any semantic information from WordNet at all. In this case the base verbs that were chosen are combined together. *I.e.* *"put down.v.1"* and *"put down.v.2"* will be grouped together. This relevancy is also treated as a baseline to compare against, and is labelled as *Action Verbs*, or AV.

Next, the notion of synonymy can be introduced and synsets can be grouped by their synonyms. This relationship within WordNet is transitive meaning that if synset A is a synonym to synset B and synset B is a synonym to synset C then A and C are also synonyms. By incorporating this information the number of classes decreases as synsets are grouped together and is designated *Action Synonyms*, or AS.

Finally, the hyponymy relationship can be used to further group synsets together. However, the hyponymy relationship can cause issues when grouping the synsets into classes. For example, the synset *"move.v.2"*, *"cause to move or shift into a new position or place, both in a concrete and in an abstract sense"* has a hyponym of *"drop.v.1"* with definition *"let fall to the ground"*. *"Move.v.2"* also has another hyponym of *"lift.v.3"*, *"move upwards"*. From these relationships *"move.v.2"* is related to both other synsets, but the

[8] a hypernym is a more generic word — the opposite relationship to a hyponym, *i.e.* *"wash.v.3"* is a hypernym of *"rinse.v.1"*.

other synsets are not related to each other, having completely separate meanings. Two classes for each synset cannot be created as this would cause *"move.v.2"* to be present in both. In order to solve this issue the lowest common subsumer is chosen as the class label and all verbs are grouped together accordingly. Note that this does mean that for the *"move.v.2"* class both *"drop.v.1"* and *"lift.v.3"* are grouped under this general verb. This semantic relevancy, denoted as *Action Hyponyms* or AH, further reduces the number of classes compared to both AS and AM.

3.5 Experiments and Results

In this section, experiments and results are presented testing the proposed method in the previous section on the annotations collected for BEOID in section 3.3. Firstly, section 3.5.1 contains implementation details and baselines that will be used for comparison. Next, evaluation using the verb meanings will be presented in section 3.5.2 before results using only the verbs will be presented in section 3.5.3 which will also compare across three different datasets: BEOID, CMU-MMAC and GTEA Gaze+.

3.5.1 Implementation and Baselines

Features Two different feature descriptors are used for all of the experiments: Improved Dense Trajectories (IDT) [133] and Overfeat CNN features [113].

- **Improved Dense Trajectories:** Videos were split into action segments before the dense trajectories were extracted. Due to the size of the features they were randomly sampled so that 25% remained (this was found to have little effect on the accuracy).

- **Overfeat Convolutional Neural Network:** The CNN was pretrained on ImageNet classes. Starting from the first frame, every 5th frame from each action segment was rescaled to 320x240 pixels and used as input. Features were taken from the penultimate layer (*i.e.* before the final FC layer).

Encoding Methods Two different schemes are also tested on top of the extracted features: Bag of Words (BoW) [21] and Fisher Vectors (FV) [111].

- **Bag of Words:** A vocabulary or codebook is created using k-means clustering over all of the different feature vectors. Codewords, *i.e.* the final representations of each video, are then created as histograms of the cluster centres. Different codebook sizes, λ_{BoW} are considered.

- **Fisher Vectors:** Similar to BoW, a codebook is created, however, this is modelled as a Gaussian Mixture Model (GMM). The representations per video thus encode first and second order statistics of the data. The size of the codebook, λ_{FV}, is also evaluated.

Additionally, Principle Component Analysis (PCA) was applied to the resultant features to further reduce their size whilst keeping the important information from the feature-encoding pair.

Baselines In all of the experiments two baselines are used to compare with SEMBED, proposed in section 3.4. These are k-Nearest Neighbour (k-NN) and Support Vector Machine (SVM).

- **K-Nearest Neighbours:** K-Nearest Neighbours is a non-parametric method in which the class of an unknown item, x_u is chosen to be the majority class of the k closest training examples to x_u. K-Nearest Neighbours is highly related to the proposed method and can be considered a special case when $t = 0$ (the number of steps, eq 3.9) in addition to the distance between neighbours not being considered, instead only the majority class is returned.

- **Support Vector Machines:** Support Vector Machines (or SVM) is a machine learning method which tries to classify data points via the construction of hyperplanes that separate the data based on class. It does this via the use of support vectors which define a margin which best split the data. Due to the use of the kernel trick it is also possible to learn non-linear decision boundaries but, considering the large number of overlaps between classes in an open vocabulary problem, this can lead to overfitting.

Implementation All experiments were implemented as a leave-one-person-out cross validation. *I.e.* for each video being used as the unknown example all videos that participant annotated were also removed from the training set to remove bias.

The baselines and the proposed method were all written in C++98 using the BOOST library for matrix multiplication and libsvm [15] to implement the SVM. The SVM was trained until the termination criterion $\epsilon = 0.001$ was met which was found to be enough for training to finish[9].

Additionally, due to the long-tailed nature of the datasets (and especially BEOID), the classes were weighted according to the following formula during the SVM training:

$$w(C) = 1/prior(C)^{\lambda} \tag{3.13}$$

where $prior(C)$ returns the prior probability of class C and $\lambda \in [0, 1]$ which is determined to be the best fit for the distribution of actions per class for a given dataset. This re-weighting was necessary in order for the model to learn both common and uncommon verbs.

Unless otherwise stated, results were gained using $k = 5$ for k-NN, $m = 240$ for SEMBED and the encoding parameters $\lambda_{BoW} = 256$ and $\lambda_{FV} = 10$ were used for Bag of Words and Fisher Vectors respectively.

3.5.2 Results on Verb Meanings

Table 3.4 shows the results of SEMBED against the two baselines, k-Nearest Neighbour and Support Vector Machines, for the four different semantic relevancies as described in section 3.4.3. The results include all combinations of CNN/IDT features and FV/BoW and all levels of relevancy AX∈{AM, AS, AH, AV}, values in brackets give the number of classes for each AX.

From the results, the combination of Improved Dense Trajectories features and Bag of Words as the encoding method consistently gives the highest results for the three different methods across all levels of semantic relevance. As the CNN features were used off the shelf with no fine-tuning it is likely that this caused the worse performance over using the IDT features.

[9] see [15] for more details.

Features	Encoding	Method	AM (108)	AS (102)	AH (84)	AV (75)
Chance			5.5	5.8	6.2	7.1
CNN	FV	SVM	13.2	17.9	18.1	20.9
		k-NN	24.6	25.6	25.0	34.4
		SEMBED	26.2	27.1	26.9	37.5
	BoW	SVM	12.1	12.7	12.2	15.2
		k-NN	7.8	8.1	7.4	19.2
		SEMBED	11.7	12.7	16.3	19.6
IDT	FV	SVM	25.9	29.8	36.2	38.7
		k-NN	28.5	30.4	33.1	36.0
		SEMBED	32.2	33.5	34.5	37.4
	BoW	SVM	26.1	29.6	29.1	34.8
		k-NN	31.6	33.6	35.2	39.6
		SEMBED	**38.2**	**40.6**	**41.9**	**45.0**

Table 3.4: *Results of SEMBED against the two baselines SVM and k-NN on BEOID using the three different semantic relevancies AX* $\in$ *{AM, AS, AH, AV}, numbers in brackets denote the number of different classes in the dataset. Results were evaluated using* $z_{CNN} = \{3, 3, 2, 4\}$, $t_{CNN} = \{20, 20, 14, 8\}$, $z_{IDT} = \{6, 10, 13, 14\}$ *and* $t_{IDT} = \{20, 20, 2, 10\}$ *for {AM, AS, AH, AV} respectively. Note that as the number of classes change between semantic relevancies, the results cannot be directly compared across columns.*

Especially of note is the poor performance of SVM in relation to the other two methods for all semantic relevancy levels AX. Due to its nature of trying to separate out the videos into classes it inherently struggles on the open vocabulary labels because of this. Both k-NN and SEMBED don't have this issue, and are shown to be able to deal with class overlaps much better. In terms of encoding, SVM performs best using IDT FV for Action Hyponyms (AH) and Action Verbs (AV), which can be attributed to how the Fisher Vector representation was created to allow simple linear classifiers to be learned on top of the representations.

The difference in number of classes between AV and Action Meanings (AM) shows the high number of variation in the number of synsets that annotators chose. By introducing semantic knowledge in the form of synonymy using the Action Synonyms (AS) relevancy, an increase in accuracy of 1.6% is seen from a reduction of 6 classes. Adding in the semantic relationship of hyponymy (AH), further reduces the number of classes much more significantly, but a similar increase in accuracy is seen, 1.3%. These accuracy increases are also mirrored in the results for both k-Nearest Neighbours and Support Vector Machines, suggesting that the grouping of classes is the cause of the accuracy increase — not the addition of semantic information. Nevertheless, SEMBED still outperforms the two baseline methods on all four levels of action relevancy.

The results using Action Verbs (AV) again reinforces the notion that semantic knowledge isn't helpful. From the table, it can be seen that simply using the verbs by themselves, and discarding the information provided by annotators using WordNet, gives the highest accuracy. Further investigation as to why this is the case brings up two examples of where annotators chose two or more different meanings for the same verb and will be explored further below:

Firstly, for the verb *"hold"*, two different synsets were chosen: *"hold.v.1"* with the definition *"keep in a certain state, position"* and *"hold.v.2"*, *"hold in one's hand"*. In this case, the synsets help differentiate between someone holding a button and someone holding/grasping an object, and would prove beneficial to the classifier.

Secondly, for the verb *"turn"*, again, two different synsets were chosen by the annotators: *"turn.v.1"*, *"change orientation or direction"* and *"turn.v.4"*, *"cause to move around or rotate"*. Note that these were used by the annotators interchangeably. Whilst both synsets have very similar meanings from their definitions (it is hard to argue which one is more correct in the case of *"turn[ing] on [a] tap"*) they are not linked in WordNet by either synonymy or hyponymy. In fact, using the Wu Palmer's distance [143] they are both more related to the verb *"close.v.1"* (with the definition *"move so that an opening or passage is obstructed; make shut"*) than each other.

Overall, 32 verbs had 2 or more synsets chosen with *"pull"* having the most at 7. Many of these were used interchangeably, including *"pull.v.10"* which had the description *"operate when rowing a boat"*: Even a synset with a very specific context within the dataset of using the rowing machine was used in multiple different contexts by annotators. Furthermore, in this and the other cases, using the semantic information within WordNet doesn't solve this issue. Three synsets ($pull.v.1$, $pull.v.2$ and $pull.v.9$) are all hypernyms of *"move.v.2"* which all concern using a force to pull an object closer. Two other synsets are hyponyms of the synset *"move.v.1"* which have definitions focused on moving/travelling. The final two pull synsets, *"pull.v.4"* (*"apply force so as to cause motion towards the source of the motion"*) and *"pull.v.5"* (*"bring, take, or pull out of a container or from under a cover"*) are not related to any of the other synsets at all. Other verbs with multiple synsets include *"take"* with 4 different synsets chosen and *"grab"* with 3. This multitude of synsets chosen per verb highlight the unsuitability of WordNet for this

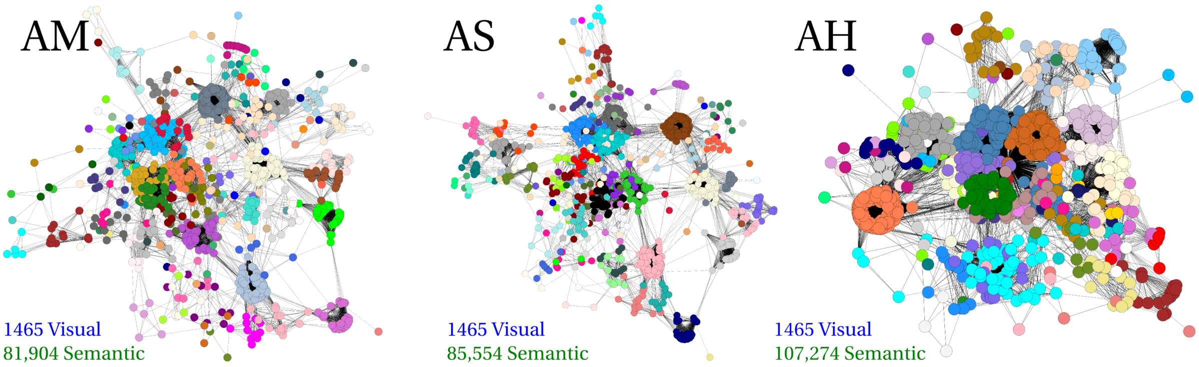


Figure 3.9: *Differences in graph structure of the trained SVG on three different levels of semantic relevancy using WordNet, Action Meanings(AM), Action Synsets(AS) and Action Hyponyms(AH). Below each graph the number of visual and semantic links are shown in blue and green respectively. Colours represent classes for each level of action relevancy.*

task, especially regarding the verb hierarchy: One would expect that many of the pull synsets described above would have some common root (hypernym), however, there are none, leading to no way of using the semantic knowledge from WordNet to link them together.

Figure 3.9 shows the impact of the semantic relevances moving from the action meanings to adding in synonymy and hyponymy relevances. The grouping of clusters in the centre (blue, orange, purple, green) of AH shows the size of the *"move.v.2"* class. Comparatively, the AM and AS graphs are much more spread out in the plot due to the lesser amount of semantic links. This leads to a large number of classes which can be semantically and visually similar compared to labels from a closed vocabulary.

A qualitative example of the benefits of the SEMBED algorithm can be seen in Fig. 3.10. It can be noted how the markov walk used in the SEMBED method is able to correct errors that might be caused by simply using k-Nearest Neighbours. Additionally, SVM, being a one-vs-all approach, struggles to deal with the amount of visual similarity between classes and predicts a class that is not connected semantically to the ground truth class, but instead has a background that is visually similar.

Verb Meaning Conclusion

This section has presented results of applying the proposed method, SEMBED, on BEOID with the collected annotations using WordNet synsets and semantic relation-

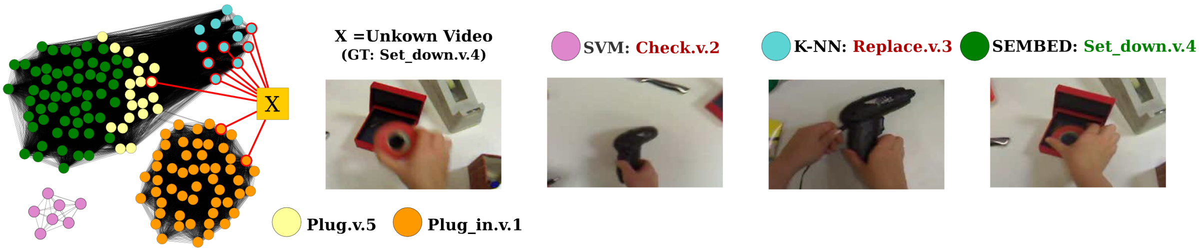


Figure 3.10: *An unknown video with the label "set down.v.4" ("cause to sit or seat or be in a settled position or place") is embedded into the graph. The z closest neighbours are videos which are visually similar ("plug.v.5", "plug in.v.1" and "replace.v.3") but incorrect semantically. By traversing the graph using a markov walk the correct class can be found.*

ships. Adding in synonymy and hyponymy links within the method allowed for some modest improvements for SEMBED, but these improvements were similarly seen for both k-Nearest Neighbour and SVM suggesting that the decrease in the number of classes caused the increase in accuracy.

This was explained via the issues with the annotation process. The annotators chose a large number of different synsets per verb which, combined with the relative sparsity of relationships within the verb hierarchy of WordNet, led videos being split into different classes of which there is no semantic relationship between them. Simply using the collected verb labels with no semantic information from WordNet gave the best results, but only visually similar verbs would be linked together within the SVG of SEMBED. This means that two videos with labels *"pull drawer"* and *"open drawer"* are treated as two distinct classes.

Other semantic knowledge bases could have been used to provide sense information for the different verbs, such as OntoNotes [50, 139, 140, 141]. OntoNotes consists of a large corpus of textual data that had been annotated with both structural and semantic information. Importantly, whilst the (verb) senses in OntoNotes are a subset of the synsets within WordNet, OntoNotes was annotated by multiple annotators with an agreement of at least 90%. It can be assumed that OntoNotes would have less ambiguity between verb senses, however the sparsity of the verb hierachy within WordNet would still be an issue. The next chapter of this thesis will look into learning relationships between verbs from context.

3.5.3 Results on Verbs

Comparison of SEMBED on Closed Vocabulary Datasets

This section presents results of SEMBED on two other egocentric datasets: CMU-MMAC [130] and GTEA Gaze+ [31]. Both datasets were originally collected by the authors with the standard verb-noun label setup: The verbs and nouns were pre-selected by the collectors and annotators had to choose the most valid word during the annotation process. Additionally, full results of the features and encoding experiments are presented as well as hyperparameter testing of z and t for SEMBED. Implementation of SEMBED and the baselines of k-NN and SVM are the same as defined in section 3.5.1. Unless specified, Action Verbs (AV) is used as the semantic relation function AX for SEMBED (see section 3.4.1 for more details).

Table 3.5 contains the results of SEMBED against the two baselines on the three different datasets. The difference in the number of classes of a closed vocabulary can immediately be seen with CMU-MMAC and GTEA Gaze+ having a total of 12 and 25 different classes respectively. Comparatively, BEOID includes 75 different classes.

On both of the closed vocabulary datasets, SEMBED underperforms, resulting in a 8% drop in accuracy compared to SVM on the smallest dataset CMU-MMAC and a 3% drop in accuracy compared to both baselines on GTEA Gaze+. It is interesting to note, that as the size of the vocabulary increases, and thus the confusion between different verbs increases due to the amount of semantic ambiguity, the difference in accuracy between SEMBED and the other two methods decreases.

SVM outperforms SEMBED on both closed vocabulary datasets, specifically beating both other methods by a large margin on CMU-MMAC. This can be explained due to its nature compared to the larger GTEA Gaze+ dataset. The actions in CMU-MMAC are unambiguous compared to those in GTEA Gaze+ with each verb being paired with only one or two nouns. In contrast GTEA Gaze+ includes many more nouns per verb leading to very similar looking actions, which can be seen in the results in which SVM performs comparatively to k-NN. For BEOID, SVM performs worse than both other methods, apart from with the combination of IDT and FV, suggesting that as the number of verbs increases the one-vs-all approach doesn't scale.

Features	Encoding	Method	CMU-MMAC (12)	GTEA Gaze+ (25)	BEOID (75)
CNN	FV	SVM	58.6	15.6	20.9
		k-NN	46.6	30.0	34.4
		SEMBED	46.3	31.0	37.5
	BoW	SVM	55.9	25.1	15.2
		k-NN	43.3	33.5	19.1
		SEMBED	52.0	33.6	19.6
IDT	FV	SVM	**69.4**	**43.6**	38.7
		k-NN	58.1	**43.4**	36.0
		SEMBED	57.4	42.1	37.4
	BoW	SVM	55.9	27.8	34.8
		k-NN	57.6	34.5	39.6
		SEMBED	61.6	40.3	**45.0**

Table 3.5: *Results of SEMBED against the two baselines SVM and k-NN on three different datasets: CMU-MMAC, GTEA Gaze+ and BEOID, numbers in brackets denote the number of different classes in each dataset. Results were evaluated using $\gamma_{fv} = 10$ and $\gamma_{BoW} = 256$, $m = 240$, $k = \{3, 5, 5\}$, $z_{CNN} = \{2, 6, 4\}$, $t_{CNN} = \{20, 20, 8\}$, $z_{IDT} = \{4, 5, 14\}$ and $t_{IDT} = \{4, 20, 10\}$ for \{CMU-MMAC, GTEA Gaze+, BEOID\} respectively. AV was used for the level of semantic relevancy for results on BEOID.*

Evaluation of Feature Extraction and Encoding Methods

Figure 3.11 shows results of varying both γ_{BoW} and γ_{FV} for both encoding methods across all three methods and datasets. For the encoding schemes, there is a strong preference for high values of γ for Bag of Words but small values of γ for Fisher Vectors. This trend is noticed across all three datasets and methods with the optimal γ_{BoW} value lying around 100 and the optimal γ_{FV} value of 10. Overall, using the combination of IDT and BoW performed best across all datasets and methods, slightly outperforming IDT and FV.

SEMBED Hyperparameter Study

Figure 3.12 shows the results of varying the hyperparameters of the SEMBED method z (number of initial neighbours) and t (number of steps in the Markov Walk). Whilst each combination of features and encodings causes large variation in optimal values of z and t, across dataset for the same pair leads to relatively similar values of z and t that

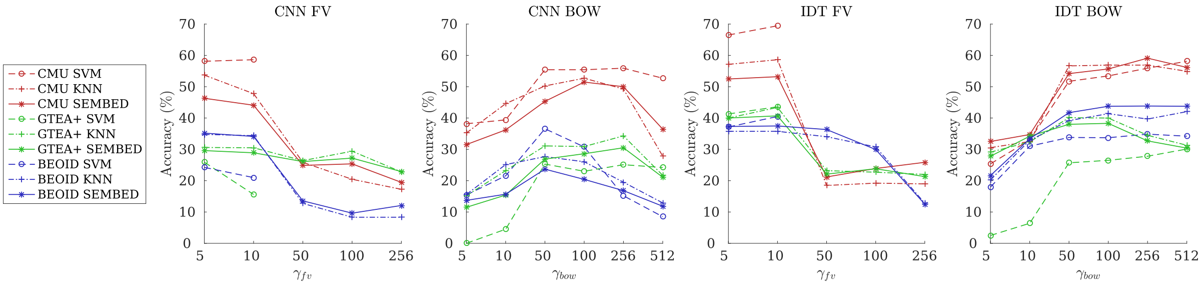


Figure 3.11: *Results of using different values for γ_{BoW} and γ_{FV}. Values were calculated with parameters $k = 5$, $m = 240m$ $z = 10$ and $t = 10$. Note that due to the size of the features for SVM $\gamma_{FV} \in \{5, 10\}$ but similar performance was seen.*

lead to the highest accuracy.

However, by looking at good values of z and — especially — t, interesting insights about the datasets can be found. In BEOID, for the semantic relevances AV, AM and AS, SEMBED is less sensitive to the values of either z or t (especially for IDT features). Conversely, for AH, lower values of t lead to optimal results. This is a clear outcome of adding in the hyponym relationship, as described in section 3.4.3, verbs start to become grouped together and therefore lower values of t are enough to 'find' the correct class in the Markov Walk. This can be seen clearly in the class *"move.v.2"* which, as mentioned in section 3.4.3, includes synsets with differing meanings such as *"pull.v.2"*, *"push.v.1"* and *"pick up.v.1"*.

The results in CMU-MMAC present a similar story, in that this dataset generally favours smaller values of t in addition to smaller values of z. Due to the nature of the closed vocabulary of CMU-MMAC, this is expected as each class is semantically distinct and, by increasing the values of z or t, it becomes likely that the method shifts the predicted class from correct to incorrect.

For GTEA Gaze+, it can be seen that the t parameter has less of an effect compared to the other datasets. Carefully choosing the z parameter is much more important which seems to be optimal around values of 4-8. This suggests that so long as the correct neighbours can be found the method is less likely to shift its focus to incorrect classes given a minimum value for t (for IDT BOW a value of $t > 8$ gives best results for SEMBED).

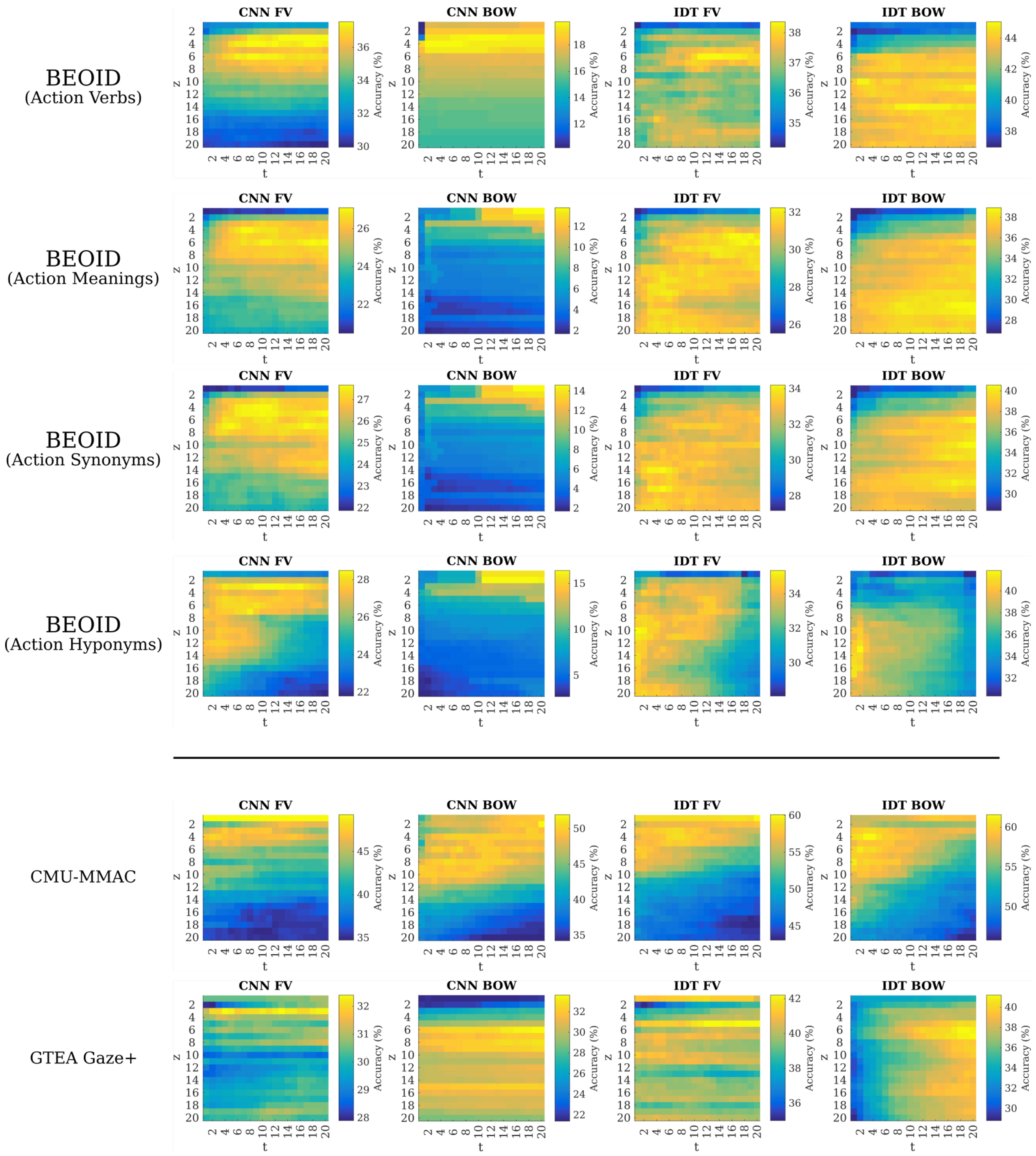


Figure 3.12: *Hyperparameter tests of z and t in the range of $[2, 20]$ for all three datasets and, for BEOID, all action relevances.*

3.6 Conclusion

In this chapter the semantic ambiguity of verbs has been introduced. Temporal annotations were collected for the BEOID dataset where annotators were allowed to freely choose which verbs to label videos. This leads to a much larger vocabulary size than what is typically seen in classical video action recognition datasets such as CMU-MMAC and GTEA Gaze+.

The larger vocabulary size causes a number of issues as the annotated verbs frequently have similar meanings and cannot be split into the non-overlapping classes that are frequently used within standard action recognition tasks. Two factors were introduced to reduce the ambiguity of verbs: Using WordNet to determine the specific meaning of each verb and to relate these meanings via relationships as well as presenting a method, SEMBED, which links verbs both semantically and visually.

SEMBED outperforms the baseline methods of k-NN (of which the proposed method is an extension) and SVM. Of particular note, the SVM classifier performs particularly poorly on the extended vocabulary due to its nature of trying to split the classes. Given the number of overlaps inherent in a large vocabulary this led to SVM underperforming in comparison to the other two methods. For closed vocabularies, such as the ground truth of CMU-MMAC and GTEA Gaze+, SVM achieves a higher accuracy because of this.

The semantic knowledge from WordNet proved to be a hindrance leading to annotators becoming confused between similar meanings of synsets. Additionally, the structure of the verb hierarchy is sparse compared to the noun hierarchy, causing many synsets to not be related even when their meanings are very similar. The combination of these issues meant that when using semantic information from WordNet the proposed method had a lower accuracy than not using any semantic knowledge at all.

This can be seen as an issue of context, where for some actions two verbs may be interchangeable but for others this may not be the case. For example, *"push"* and *"open"* are interchangeable for *"open door"* but not for *"open jar"*. These relationships would be hard to discover within semantic knowledge bases. Given the multiple annotations collected per video for BEOID, overlapping video segments could be explored to provide a measure of semantic relevancy, but due to the variability of start/end times this could lead to noisy results. The next chapter will explore the contextual relationships between verbs further, where annotations will be collected specifically to overcome this issue.

Chapter 4

Learning Semantic Information from Context

This chapter focuses on dealing with the ambiguity of verbs when used for the task of object interaction recognition. Primarily, this is done by introducing multi-verb labels which can be used to discriminate between actions without the use of nouns. Two such labelling methods are proposed, using soft and hard assigned labels respectively, which have benefits for both action recognition and the task of action retrieval, introduced in this chapter.

The previous chapter (chapter 3) introduced issues that become apparent when a video is labelled by a single verb and when the vocabulary of verbs used for action recognition is expanded. Notably, these are caused by the significant overlaps between verbs which can be interchangeable in certain contexts but not in others. This chapter looks further into the different types of verbs that can be used to describe an object interaction and the relationships between verbs.

The inclusion of multiple verb types allows for the creation of verb-only representations. Whereas a single verb label leads to ambiguity in the action described, the presence of multiple verbs can be used to fully describe an object interaction. A verb-only representation also allows a further exploration into using an expanded vocabulary in the task of action recognition which is a necessity for the task of action retrieval. The verb-only representation is collected for three different datasets using an expanded vocabulary than the original labels.

Specifically, details of the different types of verbs that exist and how they are related when used to describe actions can be found in section 4.2, information on why semantic knowledge bases cannot be used for the automatic discovery of related verbs is detailed in section 4.3. Then different verb-only representations are presented in section 4.4 before details of their collection in section 4.5 and learning in section 4.7. Finally, experiments and results can be found in section 4.8.

4.1 Moving Towards an Object-Agnostic Labelling

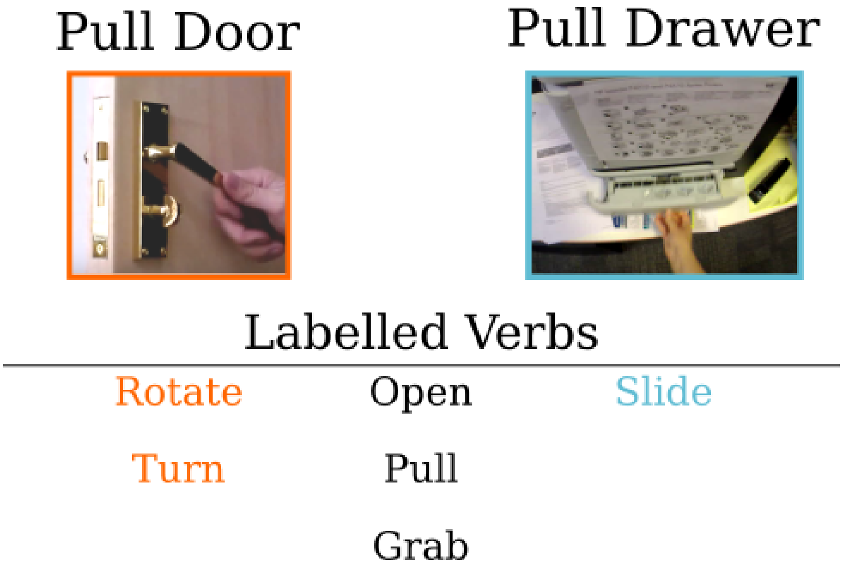


Figure 4.1: *Examples of labelled verbs for two "open" videos. Both videos share the three verb labels of "open", "Pull" and "Grab". However, they can be uniquely identified via the presence of "Rotate" for "Open Door" and "Slide" for "Open Drawer".*

In chapter 3 multiple annotators chose different verbs in order to describe an action. For the simple action of opening a drawer annotators gave 4 different words (*"pull"*, *"slide"*, *"grab"* and *"open"*, see Figure 4.1.). All of these verbs can be used to describe the action, although *"grab"* would only depict the part of grabbing the handle and not the full action. The combination of these verbs are interesting: On their own, each verb can be interpreted as belonging to many different actions but together they provide more detail as to the motion of the action. Compare this to pulling open a door, both *"pull"* and *"open"* would still be used, but the addition of *"rotate"* or *"turn"* for turning the door handle and *"slide"* for the drawer would differentiate the two actions (Figure 4.1). Therefore, any action can be defined as a set of different verbs offering the chance for a verb-only representation of actions.

This diverges from standard approaches which use a combination of a single verb and a single noun to differentiate an action. Indeed, Sigurdsson *et al.* [117] find from a study on the charades dataset that using only a single verb without any accompanying nouns caused confusion amongst annotators. However, the use of nouns forces the action to be tied to instances of objects. That is, by describing an action as *"pick-up plate"* a very similar action both visually and semantically, such as *"pick-up bowl"*, is treated as a completely separate class when only the object is different. This can force methods to learn a less generalisable model.

Additionally, this has another unforeseen effect of grouping different actions on the same object for the same goal together. For example, doors can be opened either by pushing or pulling. The verb to describe the action is the same, as is the object, however the motion is different. This can also be seen in other actions and objects such as (light) switches which are pressed/flipped/rotated or even taps which can be rotated/pressed. A large benefit of this is the ability to distinguish affordances of objects, of particular use for robot interaction.

Action Retrieval

The object-agnostic/verb-only representation also allows for exploration into another task other than action recognition, that of action retrieval. With multiple verbs being present in the representation, in addition to an order of verbs which are more relevant than others (*e.g* in the example above of opening a door *"open"* is clearly more important to describe the action than *"grab"*). These two factors, the multi-label nature of the problem and the ability of ranking verbs means that the verb retrieval task can be used to evaluate the multi-verb representations.

Formally, the retrieval task can be presented as follows: Given a query item, the task is to retrieve all items, in order of their relevancy, to the query item. This can be binary in that retrieved items are either relevant/irrelevant or each item can have a relevancy score. For the multi-verb representation specifically, a video represents a query with the aim of retrieving the relevant verbs in order of relevance. Therefore, for the *"open door"* video an example ranking might be *"open"*, *"pull"*, *"grab"*, *"rotate"*, *"turn" etc.* As the query item is a video and the retrieved items are textual in nature this is given the name video-to-text retrieval.

Additionally, as videos can be described by one or more verbs so too can verbs be used to describe one more videos and, accordingly, text-to-video retrieval can also be performed. In this case, the query item is a verb and the items to be retrieved are videos in which that verb is relevant.

Although using an expanded or open vocabulary can cause issues for the task of action recognition, mainly due to the requirement of discriminating and classifying videos into certain actions, this is not present when evaluated for action retrieval. As the items to be retrieved are ranked according to their relevancy there is no problem in having multiple relevant verbs or even verbs with differing levels of relevancy. Because of this, retrieval is highlighted here as an important task to evaluate on when using an open vocabulary.

4.2 Types of Verbs

This section introduces the types of verbs that can be used to describe actions, using the definitions of Manner verbs and Result verbs from linguistics, before presenting relationships between the verbs present in a verb-only representation.

Given an example of a person opening a push-door there are many verbs that could be used to describe the object interaction. *“Open”* is clearly one such verb, but other verbs also exist. *“Push”* gives the direction the door opens but one can go further to describe the interaction by describing also the motion that is used to turn the handle using verbs such as *“grab”*, *“hold”* and *“turn”*. This sub-action, called such as it doesn’t define the full action that takes place, is still necessary to be completed but is often ignored.

4.2.1 Manner and Result Verbs

As discussed in the previous section, many verbs can be used to describe the same action, however they all rely on the action to be given context. For example, when opening a bottle *“open”* and *“twist”* can both be used to describe the same action, whereas for opening a door *“open”* and *“push”* are instead seen as complementary in describing the action. These verbs aren’t related semantically, they are neither synonyms or hyponyms, but there exists some relationship between them for when they can be used together and when they can not.

In linguistics, many works [6, 12, 35, 37, 45] describe verbs as being either *result* or *manner* verbs (introduced in section 2.1.5):

- A result verb specifies the change in state of the action and its resulting state. *"Open"* is the result verb for the open door example.

- A manner verb specifies the manner in which the action is carried out, *i.e.* the motion required. *E.g.* in the earlier example, *"Push"* is the manner verb.

As discussed previously, Manner and Result verbs are related through context of the action not semantically and as such, their relationships can be hard to discover in semantic knowledge bases (see section 4.3.

4.2.2 Verb Importance and Relationships

The term *main verb* is introduced here to label verbs which describe the main part of the action, these could be used solely to depict the full object interaction. In any case, a main verb can be either a manner verb or a result verb. *"Open"* would be an example of a main result verb for the action of open door. An action could have multiple main verbs, but it should always have more than one.

The sub-action of an action also contains useful information about how the action is performed or steps required to complete it. To describe these, the term *Supplementary Verb* is introduced to name such verbs. These supplementary verbs don't fully describe the action, but examples are still either manner or result verbs describing either the motion or end state of the action. Actions will generally have one or more supplementary verbs.

Whilst it is possible that main verbs can be related semantically, *e.g.* *"put"* and *"place"*, this is only common between main verbs which are either both manner or both result, due to manner verbs and result verbs not being synonyms. For differing types of verbs they are instead related via context. This is because the verb types describe very different aspects of the action and therefore wouldn't be semantically related. For example, *"open"* is not related semantically to the manner verbs *"pull"*, *"push"* or *"twist"*, but *"access"* or *"unzip"* are related semantically.

Supplementary verbs follow similarly to the main verbs as discussed above. They can be

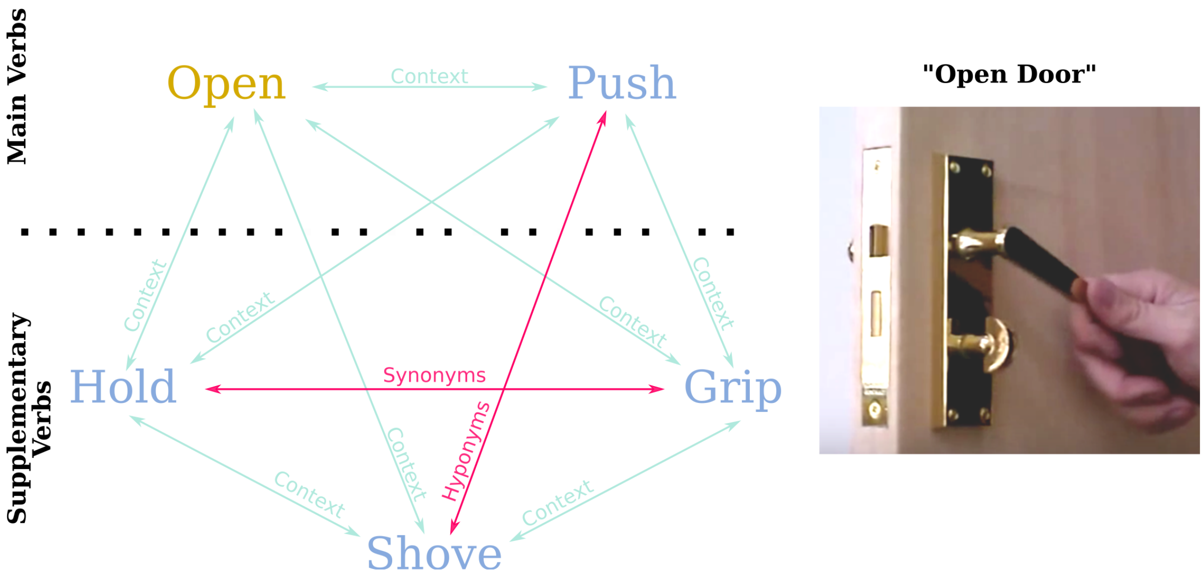


Figure 4.2: *The relationships between different verbs that can be used to describe an action ("open door" in the photo). Manner and result verbs are labelled in blue/yellow respectively. Whilst there are examples of semantic relationships being present via synonymy and hyponymy the vast majority of verbs are related through context which are difficult to find from semantic knowledge bases.*

related to each other through context, *i.e.* *"hold"* and *"turn"*, but others can indeed be found by being semantically related, *e.g.* *"hold"* and *"grip"* are related via hypernymy as they are both manner verbs.

In conclusion, the relationships between main verbs and supplementary verbs are almost exclusively found through context though some can be found via the semantic relationship of hyponymy. If a main verb and a supplementary verb were synonymous then either both verbs should be classed as a main verb or a supplementary as they have the same meaning.

This is summed up in Fig. 4.2 showing the video of someone opening a door by pushing it. In this case there are two main verbs: *"open"* (result verb) and *"push"* (manner verb). These are not related semantically, only via context, as it is possible to open objects without pushing them and vice versa. The figure also shows three supplementary verbs that could be used to help describe the action, *"hold"*, *"grip"* and *"shove"*. With these examples semantic relationships are seen in the form of synonymy and hyponymy but the vast majority of relationships between verbs are still related through context alone.

4.3 Gaining Context from Semantic Knowledge Bases

As explained in the previous section, verbs are largely related via context, a currently underexplored relationship in lexical databases or corpora. This section introduces two forms of semantic knowledge bases that are commonly used in computer vision and discussing their ability to be used for a verb-only label representation. More information about both WordNet and Word2Vec can be found in sections 2.1.1 and 2.1.2 respectively.

WordNet Whilst the noun hierarchy is often used for objects in computer vision, the verb hierarchy can similarly be constructed (see section 2.1.1). However, using the verb hierarchy presents two main issues: Firstly, that the synset for each verb is required to be able to successfully use the relationships presented above. Synsets with the same lemma, *e.g.* *"pull.v.1"* (cause to move by pulling), *"pull.v.12"* (tear or be torn violently) and *"pull.v.14"* (strip of feathers) have vastly different meanings. Secondly, the verb hierarchy is much shallower than the noun hierarchy (average synset depth of 2.53 vs. 8.15) with a higher number of synsets per lemma (1.28 noun synsets per lemma vs. 2.19) leading to a sparser and shallower set of unconnected trees.

The combination of these factors cause the verb hierarchy to be difficult to use for action recognition purposes, as shown in the previous chapter, because if the choice of synset is wrong then it is unlikely that any meaningful synsets will be linked either by synonymy or hyponymy due to its relative sparsity.

Word2Vec Word2Vec as an unsupervised method has a number of issues for using it as a semantic knowledge base for relating verbs. These stem from using co-occurrences as the learning objective for similarity. Nouns which are commonly interchanged in the text have a high similarity, *e.g.* *"doors"* and *"cupboards"* can both be opened and thus the learned vectors for both words have a high similarity. However, for fine grained action recognition, a door can both be *"open[ed]"* and *"close[d]"*. Word2Vec would try to make the representation of the two words similar when they represent antonyms as they co-occur with the same objects. Note that this is due to the inherent differences of co-occurrences of verbs and nouns.

Another issue that becomes prominent with the unsupervised learning is that, as corpora used to train word embeddings aren't labelled with each word's part of speech, there is

Verb Relationships		**WordNet**	**Word2Vec**
synonyms/hyponyms	(e.g. *"push"*- *"shove"*)	✓	×
result-manner(context)	(e.g. *"open"*- *"push"*)	×	×
main-supplementary(context)	(e.g. *"open"*- *"hold"*)	×	×

Table 4.1: *Types of verb relationships and their presence in WordNet and Word2Vec.*

no way to differentiate *"press"* the noun and *"press"* the verb causing further confusion. This leads to cases where *"press"* and *"push"* not being similar, but *"push"* is similar to *"pull"*[1].

Conclusion Table 4.1 shows the different verb relationships that were discussed and whether they can be found in the two sources of semantic knowledge presented. Note that WordNet can only find verbs related by synonymy and hyponymy whereas Word2Vec doesn't model any of the relationships.

In conclusion, both WordNet and Word2Vec cannot be used alone to find relationships in a multiple verb representation. This is due to the large number of contextual relationships between both main and supplementary verbs which are absent (see Fig. 4.2). Because of this the semantic knowledge bases are instead used as complement terms in the loss function, details of which can be seen in section 4.7.1.

4.4 Representations of Visual Actions using Verbs

A verb-only representation can include singular verbs, as in the previous chapter, or multiple verbs as discussed in section 4.1. This section introduces four different types of verb representations that will be created from the annotations collected in section 4.5 and be used for the experiments in section 4.8.

Figure 4.3 shows an overview of the different verb-only representations using a simplified set of verbs.

[1] This can somewhat be alleviated by running the corpus through a part-of-speech parser before training, but generally require low levels of noise and correct sentence structures within the corpus.

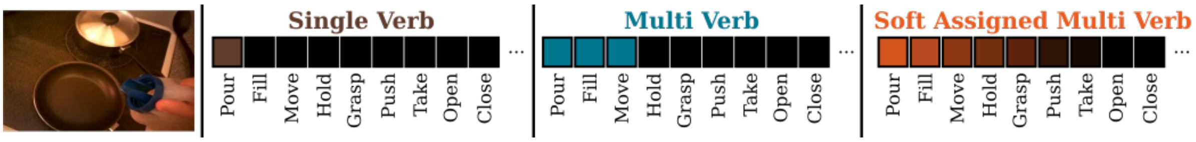


Figure 4.3: *Comparisons of different verb-only labelling representations for a video from GTEA Gaze+ with the verb-noun ground truth of "pour oil".*

Terminology To begin, the standard terminology of action recognition is presented: Given a set of X videos where x_i represents the ith video in X. Each video is given a class label y_i from the set of class labels Y.

Note that for this definition the set of Y typically contains verb-noun classes, each encoded as a one-hot vector of size $|C|$ where C is the set of all verb-noun classes. In this chapter, verb only representations are used and so the terminology is updated accordingly:

Given a set of verbs V, where the jth verb is given by v_j, each video $x_i \in X$ can be defined as having a corresponding label $\boldsymbol{y_i} \in Y$, where $\boldsymbol{y_i}$ is a vector with length $|V|$. Here, the term $y_{i,j}$ is used to represent the jth verb in $\boldsymbol{y_i}$. Note that y_i isn't a one-hot vector in the multi-label case, see below.

Single-Verb Label (SV) The naïve verb-only representation involves labelling each video with only a single verb. The label for a video, $\boldsymbol{y_i}$, is therefore of length $|V|$ and is represented as a one-hot vector. From the example in Fig 4.3, the verbs *"pour"* and *"fill"* are both relevant, but only one can be used in this labelling representation, forcing a singular verb description when multiple verbs can interchangeably be used. A more general verb, such as *"move"*, is also avoided as this would further add ambiguity into the complex space of verbs.

Verb-Noun Label (VN) The standard representation that is used for action recognition is to use the combination of a noun and a verb to disambiguate and specify the action. In this sense it is clear that the *"open"* in the action *"open door"* is different from the *"open"* in *"open jar"*. This labelling representation is used as a baseline.

In order to define $\boldsymbol{y_i}$, a set of nouns, N, is used (similarly to verbs, the kth noun is given by n_k). $\boldsymbol{y_i}$ is therefore of maximum length $|V|.|N|$ though in practice, as not every combination of verb and noun is likely to exist as an action (*e.g.* open table), this is reduced to be the size of actions in the dataset. In using this labelling representation, it is normal for verbs with multiple meanings to be avoided as well as the verb vocabulary being constructed such that there are minimal overlaps between verbs — similar to the single verb representation.

Multi-Verb Label (MV) A standard multi-labelling approach can be used where $\boldsymbol{y_i}$ is modelled as a binary vector allowing for multiple verbs to describe each video, *i.e.* hard assignment of verb labels to actions. In the example above, the added flexibility of allowing more than one verb causes both *"pour"* and *"fill"* to be descriptors of the action[2].

However, including supplementary verbs in this representation leads to problems due to the binary construction. For example, *"hold"* and *"grasp"* represent valid labels for the action, they both occur at some point during the action, yet they couldn't be used to solely describe the action. This would cause confusion between videos for which a verb constitutes the main part of the action for one video and the sub-action of another (*e.g. "hold"* in *"pour mixture"* and *"hold"* in *"hold button"*). Because of this, the relationships between main and supplementary verbs cannot be explored and, as a result, the size of the vocabulary of verbs V is limited to verbs which can be main verbs.

Soft-Assigned Multi-Verb Label (SAMV) In order to allow for both main verbs and supplementary verbs, a soft-assignment approach is used. For each verb in the label vector, $\boldsymbol{y_i}$, a numerical value between 0 and 1 is assigned which determines how applicable a verb is to describing the action. Due to this, main verbs can be given high scores whereas, comparatively, supplementary verbs will be given low scores. Formally, two verb scores $y_{i,j} > y_{i,k} > 0$ (representing scores of verbs v_j and v_k for the ith video) are assigned such values when v_i is more relevant to the action than v_k but the latter is still valid.

This is shown in Fig. 4.3 where the main verbs (*"pour"*, *"fill"*) have been assigned a larger score than the supplementary verbs (*"hold"*, *"grasp"*). Due to the restrictions of the previous representations being removed, this allows for the the set of verbs, V, to be larger and, arguably, allow for a full open vocabulary of verbs to be used in the

[2] The issue of importance is the key underlying implementation issue of using this representation. From annotations it can be difficult as to to determine what a main verb is for each action (and so find what is a relevant to use within this representation).

labelling. Additionally, by using soft assignment, verbs can therefore be ranked by their importance to an action and, because of this, the representation can be used to perform action retrieval.

4.5 Annotation Procedure

This section details the annotation procedure in which the annotations were collected to create the different verb-only representations discussed in the previous section. The three datasets from chapter 3 are used, namely BEOID, CMU-MMAC and GTEA Gaze+.

The vocabulary of verbs, V, was first constructed by combining the unique verbs from all three datasets, including the action labels collected in chapter 3, giving a list of 93 different verbs. The decision was made to exclude the verbs *"rest"*, *"read"* and *"walk"* due to the focus on fine-grained object interactions as opposed to other actions. The full list of verbs can be seen in Table 4.2. In this case, the verbs have been manually split into manner and result verbs, but this was only used for evaluation (annotators weren't given information as to what is a manner verb or result verb). It can be noted that certain examples, such as *"put"*, stray the line between manner and result due to the inherent bias of what *"put"* actions look like/involve. Additionally, somewhat interestingly, the addition of a preposition (such as *"up"* or *"down"*) often causes a verb to move from manner to result or vice versa. This can be seen in the case of *"put down"* or *"turn on"* of which the former gives a motion to the result verb and the latter describes a state of the manner verb.

Amazon Mechanical Turk (AMT) was used for the annotation process. In order for the creation of the soft assignment scores for SAMV, as well as ensuring the noise of the collected annotations was minimal, multiple annotators would be required. Due to the size of the datasets (732 for BEOID, 404 for CMU-MMAC and 1001 for GTEA Gaze+) annotating each video with multiple verbs by multiple annotators would have constituted a considerable effort. To counter this, the decision was made to annotate only a single video per original class (of which there were 95 across all three datasets) and apply the same verb labels to all videos within its class with the assumption that all videos within the original class contain the same action and therefore the verbs chosen for the single video would apply to all. Examples of this can be seen in Figure 4.4 where representative videos have been chosen for three different classes, note the intra-class similarity of the

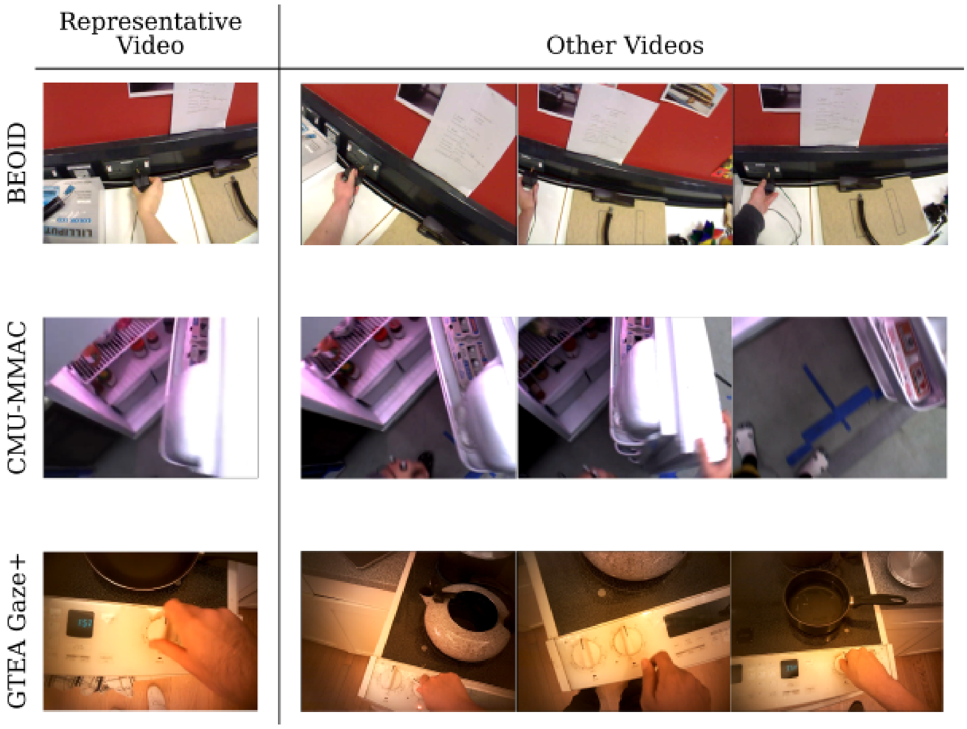


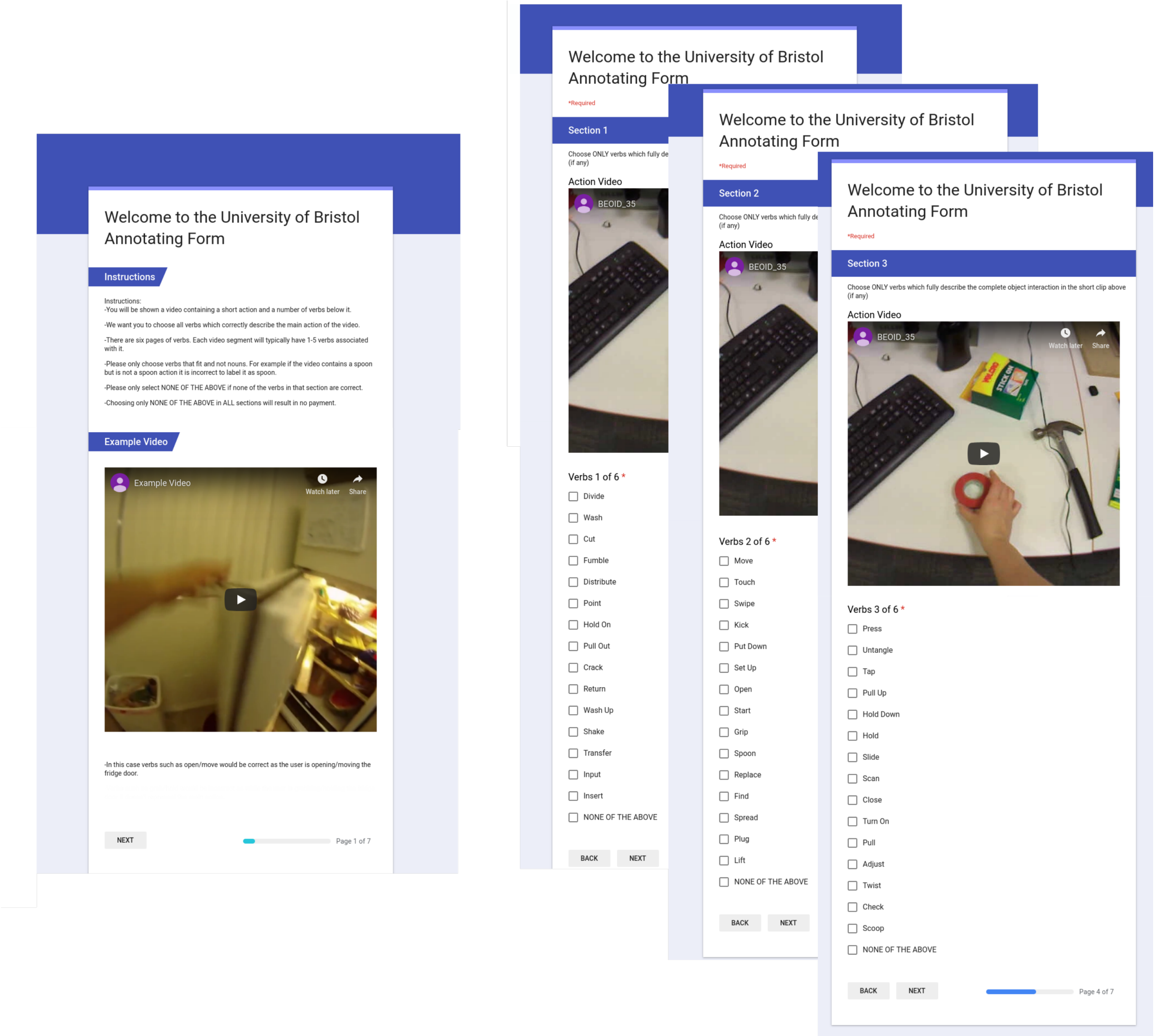
Figure 4.4: *Representative videos were chosen such that the object interaction was clear in the frame. Note that the actions taking place are in the same location for all three datasets using the same objects. Because of this, the assumption that a single video could be annotated and its labels applied to all other videos within the class holds. For other datasets, such as EPIC-Kitchens, the variation within classes would render this assumption invalid.*

videos.

Figure 4.5 shows images of the annotator that the AMT workers used. The list of 90 verbs were broken down into 6 groups of fifteen to give a trade-off between (reducing) the number of pages to click-through and (reducing) the number of verbs on screen at a single time. Trials found that more than 15 verbs could cause annotators to skip-through or miss important verbs.

The representative video per class was chosen carefully to ensure that the action being performed was visible on-screen as well as good lighting was available in the scene. This was done to further reduce noise as well as ensure that the action taking place was clear so annotators only had to decide what verbs applied and not what the action actually entailed.

The annotators were given the following as an instruction upon loading up the annotation form:



Welcome to the University of Bristol Annotating Form

Instructions

Instructions:
-You will be shown a video containing a short action and a number of verbs below it.

-We want you to choose all verbs which correctly describe the main action of the video.

-There are six pages of verbs. Each video segment will typically have 1-5 verbs associated with it.

-Please only choose verbs that fit and not nouns. For example if the video contains a spoon but is not a spoon action it is incorrect to label it as spoon.

-Please only select NONE OF THE ABOVE if none of the verbs in that section are correct.

-Choosing only NONE OF THE ABOVE in ALL sections will result in no payment.

Example Video

-In this case verbs such as open/move would be correct as the user is opening/moving the fridge door.

Welcome to the University of Bristol Annotating Form

*Required

Section 1

Choose ONLY verbs which fully de
(if any)

Action Video

Verbs 1 of 6 *

□ Divide
□ Wash
□ Cut
□ Fumble
□ Distribute
□ Point
□ Hold On
□ Pull Out
□ Crack
□ Return
□ Wash Up
□ Shake
□ Transfer
□ Input
□ Insert
□ NONE OF THE ABOVE

Welcome to the University of Bristol Annotating Form

*Required

Section 2

Choose ONLY verbs which fully de
(if any)

Action Video

Verbs 2 of 6 *

□ Move
□ Touch
□ Swipe
□ Kick
□ Put Down
□ Set Up
□ Open
□ Start
□ Grip
□ Spoon
□ Replace
□ Find
□ Spread
□ Plug
□ Lift
□ NONE OF THE ABOVE

Welcome to the University of Bristol Annotating Form

*Required

Section 3

Choose ONLY verbs which fully describe the complete object interaction in the short clip above (if any)

Action Video

Verbs 3 of 6 *

□ Press
□ Untangle
□ Tap
□ Pull Up
□ Hold Down
□ Hold
□ Slide
□ Scan
□ Close
□ Turn On
□ Pull
□ Adjust
□ Twist
□ Check
□ Scoop
□ NONE OF THE ABOVE

Figure 4.5: *The annotating form that was used for Amazon Mechanical Turk (AMT) workers. The first page gave instructions with an example video from the ADL dataset (not used in the collection) the following 6 pages included the video to be annotated each with 15 different verbs. Each page also had the ability for annotators to choose NONE OF THE ABOVE if they felt that none of the 15 verbs on that page applied to the action.*

Instructions:

-You will be shown a video containing a short action and a number of verbs below it.

-We want you to choose all verbs which correctly describe the main action of the video.

-There are six pages of verbs. Each video segment will typically have 1-5

> *verbs associated with it.*
>
> *-Please only choose verbs that fit and not nouns. For example if the video contains a spoon but is not a spoon action it is incorrect to label it as spoon.*
>
> *-Please only select NONE OF THE ABOVE if none of the verbs in that section are correct.*
>
> *-Choosing only NONE OF THE ABOVE in ALL sections will result in no payment.*

Additionally, the a video from the Activities in Daily Living (ADL) dataset was used as an example with the following text:

> *-In this case verbs such as open/move would be correct as the user is opening/moving the fridge door.*

In total $2,939$ annotators took part in the annotation process with a minimum of 30 annotators per video and a maximum of 50. Annotators were rejected if they tried to game the system by choosing 0 verbs: From looking at collected examples, as long as one verb was chosen by the annotators it was found to be relevant to the action and didn't constitute a noisy annotation.

After collection the verb scores were normalised by dividing the number of annotators that chose a particular verb by the number of annotators which annotated that action. Using $\boldsymbol{t_i}$ to represent the collected, normalised annotations for the *ith* video, x_i and $t_{i,j}$ to represent the score of *jth* verb for x_i each of the label representations introduced in section 4.4 can be constructed using the following:

- **SV** The majority vote was used to assign each video a single verb. *I.e.* $\forall i,\ y_{i,j} = 1$ if $j = \arg\max(\boldsymbol{t_i})$ and 0 otherwise.

- **VN** The verb was found using majority vote as in SV and the noun from the original class was added (from the original annotations of BEOID[3], CMU-MMAC and GTEA Gaze+). The vector $\boldsymbol{y_i}$ is then constructed as a one-hot vector where the *jth* element is set to 1, corresponding to the verb-noun pair.

- **MV** A threshold of 0.5 was used to differentiate main verbs from supplementary verbs. $\boldsymbol{y_i}$ was then constructed over all i using: $y_{i,j} = 1$ if $t_{i,j} > 0.5$ and 0 otherwise.

[3] The annotations from the single annotator who annotated every video were used.

Manner	Result
carry, compress, drain, flip, fumble, grab, grasp, grip, hold, hold down, hold on, kick, let go, lift, pedal, pick up, point, position, pour, press, press down, pull, pull out, pull up, push, put down, release, rinse, reach, rotate, scoop, screw, shake, slide, spoon, spray, squeeze, stir, swipe, swirl, switch, take, tap, tip, touch, twist, turn	activate, adjust, check, clean, close, crack, cut, distribute, divide, drive, dry, examine, fill, fill up, find, input, insert, mix, move, open, peel, place, plug, plug in, put, relax, remove, replace, return, scan, set up, spread, start, step on, switch on, transfer, turn off, turn on, unlock, untangle, wash, wash up, weaken

Table 4.2: *Manual split of the verbs from BEOID, CMU-MMAC and GTEA Gaze+ into manner verbs and result verbs.*

- **SAMV** The normalised scores were used directly as the soft-assignment scores. *i.e.* $\forall i, j \; y_{i,j} = t_{i,j}$.

After collection, due to lack of knowledge base, the 90 verbs were manually split into manner and result verbs which can be seen in Table 4.2. Out of the 90 verbs there is a relatively even split with 47 manner verbs and 43 result verbs.

Examples of the result of the annotation process can be seen in Fig. 4.6 for two videos: *“Plug in Screwdriver”* and *“Close Freezer”*. The agreement between annotators can be used to discover what verbs are considered relevant, and thus describe the main action with high assignment scores. Irrelevant verbs can also be found from the agreement between annotators as the verbs were *agreed to have a low score*, *i.e.* few to no annotators chose the verb for the action. Supplementary verbs are those in which the annotators disagree upon the relevancy, and as such, have middling assignment scores: By themselves they do not fully describe the interaction yet they cannot be argued to be completely irrelevant.

4.6 Annotation Statistics

This section presents statistics of the collected annotations from the previous section across the three datasets. Figure 4.7 includes the main statistics with (a) providing the frequency of each verb being chosen across all three datasets. In (b) the number of verbs chosen by an annotator per original class is given across all three datasets. (c) shows the number of unique verbs per class that were chosen. (d) gives the average soft assignment score for every verb showing the maximums and minimums of the assignment score given to each verb. Finally, (e) presents the top co-occurrences between verbs that

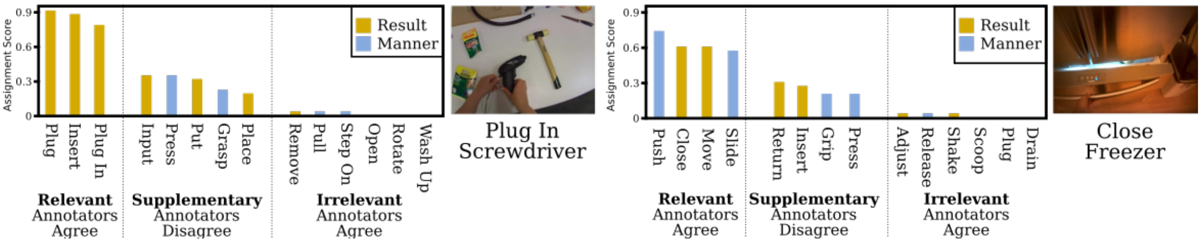


Figure 4.6: *The agreement and disagreement between annotators is used to determine what verbs describe the main action, which are supplementary verbs and which are irrelevant.*

were annotated.

Verb Frequency (Figure 4.7(a)) The verb frequency shows a long-tail distribution, with *"move"* being the most common appearing in all three datasets with a similar frequency. This figure highlights the difference between BEOID and the other two datasets in that it has scenes which aren't in the kitchen, therefore requiring different verbs than the other datasets to label the unique actions. *E.g.* *"activate"*, *"switch"*, *"plug"* and *"step on"* are almost entirely seen in BEOID.

Verbs Chosen per Annotator (Figure 4.7(b)) Shorter actions, such as *"close fridge"* and *"open microwave"*, were annotated with lower number of verbs (median of 5/6 resp.) whereas actions which included more sub-actions, such as *"pour baking pan [into] bowl"* and *"take pam [from cupboard]"*, had a median of 10 verbs chosen per annotator. From the figure, the CMU-MMAC dataset has some of the highest number of verbs chosen with, 9 out of the top 10 videos belonging to this dataset.

This is well explained from the construction of the dataset, in which the actions were labelled contiguously with no gaps between actions. This causes the videos to contain actions of a coarser grain nature. *E.g.* the action *"take scissors"* includes the action of opening and closing the drawer where the scissors reside. Moltisanti *et al.* [90] show that out of the three datasets, CMU videos have both a higher average length and a higher total length.

Unique Verbs (Figure 4.7(c)) Similar to (b), the shorter actions of *"open microwave"* and *"close fridge"* had the lowest number of unique verbs chosen per class at 19 and 20 respectively. However, the actions with the highest number of unique verbs chosen

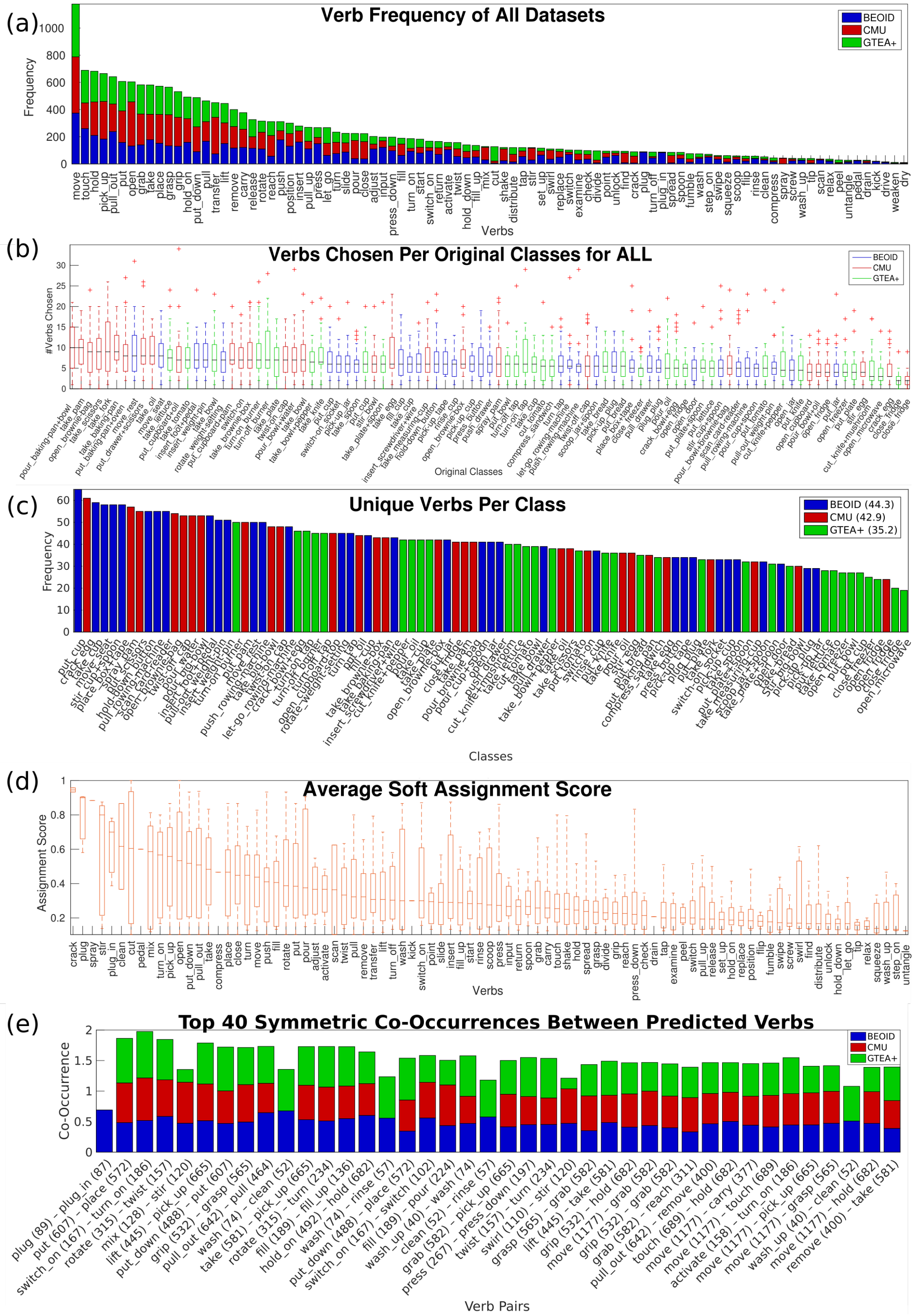


Figure 4.7: *Annotation statistics of the collected annotations. (a) The frequency of the verbs as chosen by the annotators across all three datasets. (b) The number of verbs chosen by each annotator, given as a boxplot, per original ground truth class. (c) The number of unique verbs per original ground truth class. (d) The average soft assignment score given to each verb after dividing by the number of annotators per video. (e) The top 40 co-occurrences of verbs in the annotations showing whether the annotators thought verbs were completely interchangeable with one another.*

don't correlate with the previous figure suggesting that for actions such as *"put cup"* and *"crack egg"* annotator disagreement was higher than others.

Average Verb Score (Figure 4.7(d)) Verbs *"crack"*, *"spray"* and *"pedal"* are examples of verbs which were annotated with strong agreement between annotators for only one or two actions. Comparatively, general verbs such as *"move"* have a much higher range of assignment scores.

Verb Co-Occurrences (Figure 4.7(e)) The top 40 co-occurrences are shown here, *i.e.* given a pair of words what is the percentage that they were both picked by an annotator. This gives a measure of whether the annotators believed any pair of verbs were interchangeable (as they would have always been picked together). Even the verbs *"plug"* and *"plug in"* aren't seen as interchangeable by the annotators. They were annotated with a similar number of instances (89 and 87 respectively) but were only chosen together 70% of the time. Similarly, no other pair of verbs are found to be completely interchangeable across all three datasets.

Main vs. Supplementary/Manner vs. Result Figure 4.6 includes two videos with example distributions showing main verbs and supplementary verbs along with manner and result verbs. It was found that the number of main verbs varied per video with an average of 3.35 ± 1.49 main verbs per video and 4.21 ± 2.62 supplementary verbs per video.

In regards to whether manner or result verbs were chosen no correlation was found with either being more important to the action, *i.e.* number of manner verbs being main verbs vs. number of result verbs being main verbs.

4.7 Learning Verb-Only Representations

This section introduces the comparative method used to learn the different labelling approaches defined in section 4.4.

Formally, a function ϕ will be learned: $\phi : \mathcal{W} \mapsto \mathbb{R}^{|V|}$ which maps a video representation $\mathcal{W}$ onto the verb labels. The predicted verb labels for the *ith* video, x_i, is given by

$\hat{\boldsymbol{y}}_{\boldsymbol{i}} = \phi(x_i)$, with $\hat{y}_{i,j}$ representing the predicted score for the jth verb of the ith video. To distinguish between the learned function of each method, $\phi_{\mathcal{X}}$ is used where $\mathcal{X} \in \{SV, VN, MV, SAMV\}$.

Single Label Learning SV and VN represent single label (SL) representations and it is typical to use the cross entropy loss over the softmax function σ:

$$L_{SL} = -\sum_i \boldsymbol{y}_{\boldsymbol{i}} \log(\sigma(\hat{\boldsymbol{y}}_{\boldsymbol{i}})) \tag{4.1}$$

Learning Multi Verb (MV) MV is represented as a binary vector, thus a sigmoid cross-entropy loss is used as is common for multi-label classification:

$$L_{MV} = -\sum_i \sum_j y_{i,j} \log(S(\hat{y}_{i,j})) + (1 - y_{i,j}) \log(1 - S(\hat{y}_{i,j})) \tag{4.2}$$

where S represents the sigmoid function.

Learning Soft Assigned Multi Verb (SAMV) With each verb being soft-assigned a value between 0 and 1 the learning ϕ_{SAMV} can be seen as a regression problem. Accordingly, ϕ_{SAMV} is learned using euclidean error:

$$L_{SAMV} = \sum_i \sqrt{(\boldsymbol{y}_{\boldsymbol{i}} - \hat{\boldsymbol{y}}_{\boldsymbol{i}})^T (\boldsymbol{y}_{\boldsymbol{i}} - \hat{\boldsymbol{y}}_{\boldsymbol{i}})} \tag{4.3}$$

Additionally, to evaluate the multi-label classification loss for the SAMV task, Eq. 4.2 is also used to train ϕ_{SAMV} (results in section 4.8.1). Note that by using Eq. 4.2 the value of L_{ML} when $y_{i,j} = \hat{y}_{i,j}$ is not necessarily zero, however, the gradient will be zero at these points. Figure 4.8 shows the values of the losses for different $y_{i,j}$ and $\hat{y}_{i,j}$ as well as the gradients.

4.7.1 Adding in Semantic Knowledge

Section 4.3 introduced two potential sources of semantic knowledge WordNet and Word2vec. It was shown that, by themselves, both sources would be unable to be used for learning

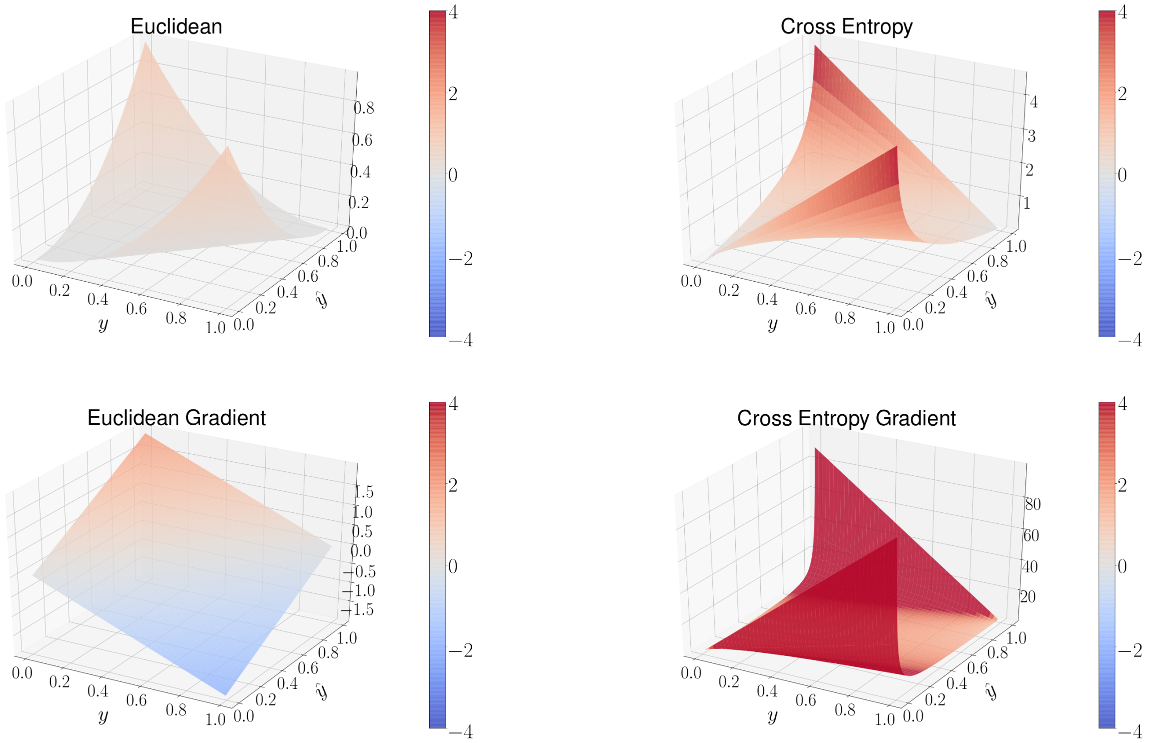


Figure 4.8: *Comparison between a euclidean loss (Eq. 4.3) and a (sigmoid) cross entropy loss (eq. 4.2) for regression. Note the line $y = \hat{y}$ is zero when considering a euclidean loss but this is not the case for cross entropy. However, the gradients of these losses when $y = \hat{y}$ are* 0.

a multi-verb representation solely but it's possible that they could still provide some benefit to the learning. This section introduces loss terms that can be added to assist the model in learning SAMV.

WordNet The WordNet synsets of the annotated verbs that were collected are not known[4] causing issues trying to find whether two verbs are synonymous or hyponymous. Even if the synsets were known, as mentioned in section 4.3, the majority of relationships in a verb-only representation are via context.

Therefore, a loss term that is constructed to penalise predictions which should be similar would fall into the trap of estranged synsets being related in WordNet. For example, in WordNet, a specific meaning of *"open"* is synonymous to specific meanings of both *"cut"* and *"move"* via the definitions of *"butterflying"* meat and making an opening (*"move"*),

[4] If they were known it is likely that there would be > 90 synsets as the verbs across datasets could have different meanings.

neither of which occur in the three datasets.

In order to provide a weakly-supervised, beneficiary loss, the loss term is instead constructed to penalise verbs which are predicted together which, according to WordNet, should never be seen together. *I.e.* for all synsets of a verb there exists no synonymy or hyponymy relationship between them.

Accordingly, for each verb, v_i, a set B_i is created which contains all verbs from the V which are antonyms[5] or are never synonyms or hyponyms regardless of verb meaning:

$$B_i = \{v_j : v_j \in ant(v_i) \wedge v_j \notin syn(v_i) \wedge v_j \notin hyp(v_i), \forall v_j \in V\} \tag{4.4}$$

where $ant(v_i)$ returns the set of antonyms to all lemmas of v_i, $syn(v_i)$ finds the set of all verbs related by synonymy to all synsets in which v_i is a lemma and $hyp(v_i)$ gets the set of all verbs related by hyponymy to all synsets in which v_i is a lemma (both hyponyms and hypernyms).

Equation 4.4 can then be used to create a loss term which punishes two verbs being predicted highly for which there is no relationship between them in WordNet.

$$L_{WN} = \sum_{i} \sum_{v_j \in V} \frac{1}{|B_j|} \sum_{k=1}^{|B_j|} \hat{y}_{i,j} \cdot \hat{y}_{i,k} \tag{4.5}$$

For example, B_{cut} = {*"input"*, *"grasp"*, *"scoop"*, *"plug in"*, *"spray"*}, which highlights the issues of using WordNet for this task. Firstly, *"cut"* and *"grasp"* are never seen as relevant, even when contextually they are common (*"grasp[ing]"* a knife while *"cut[ting]"*). Secondly, the WordNet hierarchy has links between *"cut"* and verbs such as *"step on"* or *"pedal"* which are not related within the context of the action recognition datasets.

Word2Vec The assumption of the co-occurrences of words in the corpus is an approximator of similarity was discussed in Section 4.3 to have issues when used to find relationships between verbs in the multi-verb labelling. Accordingly, it is decided to

[5] The number of antonyms in the verb hierarchy of WordNet is low, with only 9.65% of verbs having antonyms in one or more of their synsets. Additionally, it is common for the antonyms to not lie within the 90 verb vocabulary.

not use a loss term which penalises verbs which aren't predicted together with a high cosine similarity (similar to the constructed loss for WordNet). Instead, the loss term is constructed such that it penalises verbs which are predicted together which have a low similarity (and so did not co-occur in the corpus).

The set D_i is created for a verb v_i such that it contains all verbs from V which have a low similarity:

$$D_i = \{v_j : sim(v_i, v_j) < \zeta, \forall v_j \in V\} \tag{4.6}$$

where $sim(v_i, v_j)$ returns the cosine similarity between the embedded word vectors of v_i and v_j and ζ is a constant used as a threshold. For example, using the Wikipedia corpora, $D_{\{put\}} = \{$ *"fumble"*, *"press"*, *"swirl"*, *"peel"*, *"pedal"*, *"dry"*$\}$ with $\zeta = 0.1$. The loss term can thus be created in a similar way as before using the following:

$$L_{W2V} = \sum_i \sum_{v_j \in V} \frac{1}{|D_j|} \sum_{k=1}^{|D_j|} \hat{y}_{i,j} \cdot \hat{y}_{i,k} \tag{4.7}$$

Final Loss The overall loss function for learning ϕ_{SAMV} using semantic knowledge (denoted by L_{SIM}) can now be formulated as follows:

$$L_{SIM} = L_{SAMV} + \beta_1 L_{WN} + \beta_2 L_{W2V} \tag{4.8}$$

where β_1 and β_2 are constants used to weight the WordNet and Word2vec losses. Note that if β_2 (respectively β_1) is set to 0 then only knowledge from WordNet (respectively Word2Vec) will be used during training.

4.8 Experiments and Results

This section presents the experiments and results of testing the different verb-only representations against the standard verb-noun representation including results for Action

Recognition (section 4.8.1) as well as for Action Retrieval (section 4.8.2). Finally, the per verb error is also evaluated (section 4.8.3).

Implementation Details $\phi \in \{SV, VN, MV, SAMV\}$ was trained as a two-stream CNN using the code from [33]. It uses two VGG-16 networks [121][6] (a spatial stream and a temporal stream). The networks were first trained on ImageNET [109] before the temporal network was modified to have a stack of 10 optical flow frames as input. The networks were then both pre-trained on the UCF101 dataset and fine-tuned on BEOID, CMU-MMAC, and GTEA Gaze+ for 100 epochs which was enough for training to converge for all four labelling representations. In each step the streams are trained individually before trained together with the late fusion described in the original paper (this was found to give a good trade-off between accuracy and time spent training/number of parameters of the model). For each of the three datasets 5 cross-fold validation was used.

The learning rate was set to 0.001 which was decayed by a factor of 10 every 10 epochs after the 20th epoch and a dropout ratio of 0.85 was applied during training. The batch size for both streams was set to 256. Additionally the weight decay was set to 0.0005. ζ was set to 0.1.

4.8.1 Action Recognition Results

This section evaluates the different labelling representations for the task of action recognition comparing the verb-only labels to the classical verb-noun (VN) approach. It first presents results for using semantic knowledge for and comparing the use of loss function for ϕ_{SAMV} before comparing the different labelling representations previously introduced in section 4.4.

Evaluation Metric

In order to compare the accuracy for the single-label and multi-label approaches the following approach is used.

[6] The networks include 5 convolutional layers each followed by a maxpool layer before 3 fully connected layers and a softmax layer.

	BEOID				CMU				GTEA+			
β_1	0	0.5	0	0.5	0	0.5	0	0.5	0	0.5	0	0.5
β_2	0	0	0.5	0.5	0	0	0.5	0.5	0	0	0.5	0.5
Accuracy	**82.5**	**82.5**	82.0	82.1	**69.1**	67.7	67.8	67.2	**70.4**	**70.4**	69.5	69.0

Table 4.3: *Comparison of using semantic knowledge from WordNet and Word2Vec. β_1 and β_2 are weighting factors of the loss terms for WordNet and Word2Vec respectively given in equation 4.8.*

Firstly, let V_i^L be the set of verbs which are ground truth for the video x_i using the labelling representation L. The size of this set, $k = |V_i^L|$ is then used to find the top k predicted verbs using the same representation, denoted by $\hat{V}_i^L$. In the case of single-label representations, *i.e.* $L \in \{SV, VN\}$, $k = 1$, however, for multi-label representations k will differ depending on the video. The accuracy using the sets V_i^L and $\hat{V}_i^L$ can then be evaluated as:

$$A(L) = \frac{1}{|X|} \sum_i \frac{|V_i^L \cap \hat{V}_i^L|}{|V_i^L|} \tag{4.9}$$

Intuitively, this gives the percentage of correctly predicted ground truth verbs for each labelling scheme. It is important for k to vary for each video as the number of verbs which are important for describing an action differ (this can be seen in Fig. 4.6 where the videos have a different number of main and supplementary verbs).

For SV and MV, the ground truth sets V_i^{SV} and V_i^{MV} are clear, it contains all verbs in which $y_{i,j} = 1$. But for SAMV, creating V_i^L isn't so obvious. Therefore, to get the ground truth for SAMV, a threshold is used, denoted by α, creating a set of verbs such that if $y_{i,j} > \alpha$ then the verb is included in V_i^{SAMV}[7]. For the experiments, unless otherwise noted, α was set to 0.3 so as to reduce noise from rarely chosen verbs. Additionally, different values of α are tested in order to evaluate the ability for ϕ_{SAMV}'s ability to predict all verbs of importance.

Using Semantic Knowledge

Table 4.3 shows the results of using WordNet and Word2Vec for learning ϕ_{SAMV} for the action recognition task. Across all three datasets setting $\beta_1 > 0$ and $\beta_2 > 0$ (equation 4.8)

[7] Note that if $\alpha = 0.5$ then $V_i^{MV} = V_i^{SAMV}$

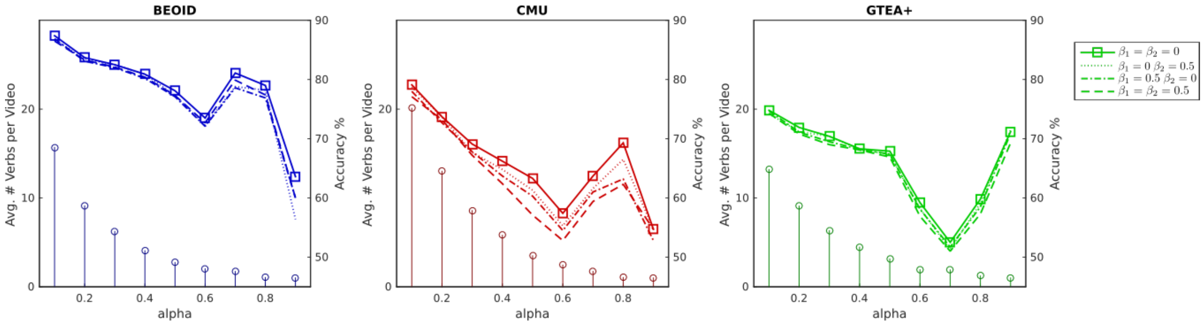


Figure 4.9: *Accuracy training ϕ_{SAMV} using euclidean loss (Eq. 4.3) compared with using semantic knowledge from WordNet and Word2Vec, trained with Eq. 4.8. Accuracy calculated using Eq. 4.9 with differing values of α.*

leads to a decrease in overall recognition accuracy performing worse than using either WordNet or Word2Vec on their own. Additionally, in all cases simply setting $\beta_1 = \beta_2 = 0$ performs comparatively or better showing their unsuitability for learning ϕ_{SAMV}.

Figure 4.9 shows how the accuracy of using WordNet and Word2Vec differs with varying levels of α (from equation 4.9). Using either of the semantic knowledge bases leads to the results being comparable or a reduction in accuracy. As highlighted in section 4.3, both WordNet and Word2Vec don't include all of the relationships present in a multi-verb representation giving reason for the drop in accuracy.

Qualitative results of using semantic knowledge can be seen in Fig. 4.10 for the largest of the three annotated datasets, GTEA Gaze+. As expected, in videos where semantic relationships are present, using both WordNet and Word2Vec offer considerable boosts to performance. For example, *"put plate"* commonly has *"put"* being confused with *"take"* however both WordNet and Word2Vec are able to predict the ground truth ranking much better. This is also seen somewhat in *"Open Fridge"* where the model predicts both *"open"* and *"close"*. Not using any semantic knowledge leads to *"close"* being predicted with a higher score (*"close"*: 0.494 vs. *"open"*: 0.399) whereas using WordNet and/or Word2Vec mitigates this somewhat by predicting *"open"* with a higher score (*"close"*: 0.349 vs. *"open"*: 0.575 when $\beta_2 = 0.5$).

Unfortunately, outside these cases, the addition of semantic knowledge struggles to help the verb prediction. For *"Compress Sandwich"*, in which in WordNet both *"press"* and *"compress"* are related through a number of synsets, it actually has an adverse effect causing *"press down"* to not be predicted at all. The addition of the semantic loss terms also causes problems with verbs that are related via context. For *"cut knife+pepper"* setting $\beta_1 = \beta_2 = 0$ correctly predicts the top 4 verbs, albeit in a slightly different order, whereas using either WordNet or Word2Vec causes these supplementary verbs to be (largely) lost.

Euclidean vs. Sigmoid Cross Entropy

Dataset	BEOID	CMU-MMAC	GTEA Gaze+
Euclidean Loss	82.5	69.1	70.4
Sigmoid Cross Entropy Loss	**87.8**	**73.5**	**72.9**

Table 4.4: *Comparison between using euclidean loss (eq. 4.3) and using sigmoid cross entropy loss (eq. 4.2) across all three datasets for learning ϕ_{SAMV}. Results shown in accuracy (using eq. 4.9).*

Table 4.4 shows the difference of using a euclidean loss and using a sigmoid cross entropy loss for learning ϕ_{SAMV}. Across all three datasets accuracy increases using the cross entropy loss seeing an increase of accuracy of 5.3%, 4.4% and 2.5% respectively. From the gradients shown in Figure 4.8, it can be seen that the Sigmoid Cross-Entropy Loss is harsher towards large disparities between y and $\hat{y}$ (ground truth and predicted respectively). Comparatively, the linear nature of the gradient of the euclidean loss gives a (relatively) higher penalty when y and $\hat{y}$ are similar. This causes the sigmoid cross-entropy loss to focus more on highly erroneous examples whilst not over-penalising examples with a small error.

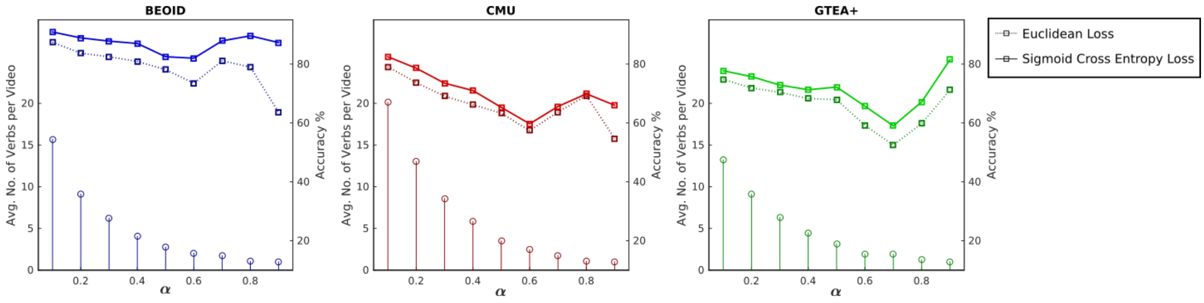


Figure 4.11: *Accuracy training ϕ_{SAMV} using a euclidean loss (Eq. 4.3) and using a sigmoid cross entropy loss (Eq. 4.2). Accuracy calculated using eq. 4.9 with differing values of α. Over all datasets and α using a sigmoid cross entropy loss to train ϕ_{SAMV} outperforms a euclidean loss.*

Figure 4.11 compares ϕ_{SAMV} being trained with a euclidean loss and a sigmoid cross entropy loss for different values of α (thus changing what is considered 'ground truth'

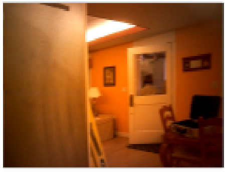
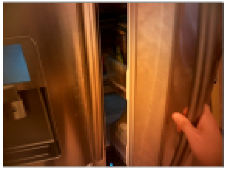
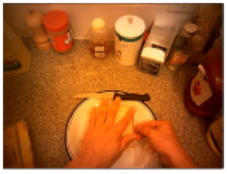
	GT	$\beta_1 = \beta_2 = 0$	$\beta_1 = 0.5\ \beta_2 = 0$	$\beta_1 = 0\ \beta_2 = 0.5$	$\beta_1 = \beta_2 = 0.5$
Put Plate 	Put Down Place Move Put	Move Take Hold Grab	Move Place Put Down Put	Move Put Down Put Place	Move Place Put Put Down
Close Fridge 	Close Push Move Flip	Move Close Put Down Put	Close Move Push Put Down	Close Move Push Put Down	Close Push Move Put Down
Put Knife 	Put Down Put Place Let Go	Move Place Put Down Put	Put Down Move Place Put	Put Down Move Place Put	Move Put Down Place Put
Open Fridge 	Open Pull Out Pull Move	Close Open Move Push	Open Close Move Pull	Open Close Move Pull	Open Move Close Pull Out
Compress Sandwich 	Press Press Down Touch Compress	Touch Press Down Grasp Move	Move Take Pull Out Grab	Move Take Pull Out Grab	Move Take Grab Pick Up
Cut Knife+Pepper 	Cut Touch Grip Hold	Cut Hold Grip Touch	Cut Move Place Put Down	Cut Move Put Down Hold	Move Cut Place Put

Figure 4.10: *Qualitative results showing the ranking of verbs given the ground truth of videos from GTEA Gaze+. Green and Red represent correct and incorrect results whereas orange denotes correct verbs which have been predicted significantly higher/lower than in the ground truth. Whilst WordNet and Word2Vecshow benefits in videos where antonyms are predicted together, e.g. "put" and "take", they struggle when relating verbs via context causing detrimental results compared to learning the SAMV representation alone.*

for SAMV). Similarly, for all three datasets, regardless of the value of α, using a sigmoid cross entropy loss results in an increase of accuracy. Due to these results the remaining results in this chapter for ϕ_{SAMV} were evaluated using the model trained with the sigmoid cross entropy loss.

Comparison of Labelling Techniques

	BEOID				CMU				GTEA+			
	ϕ_{SV}	ϕ_{VN}	ϕ_{MV}	ϕ_{SAMV}	ϕ_{SV}	ϕ_{VN}	ϕ_{MV}	ϕ_{SAMV}	ϕ_{SV}	ϕ_{VN}	ϕ_{MV}	ϕ_{SAMV}
No. of verbs*	20	40	42	90	12	29	31	89	15	63	34	90
Accuracy	78.1	**93.5**	**93.0**	87.8	59.2	**76.0**	74.1	73.5	59.2	61.2	71.7	**72.9**

Table 4.5: *Action recognition accuracy results for $\phi_{\{SV,VN,MV,SAMV\}}$ using Eq. 4.9. Additionally, number of verbs that each method learns is given (for ϕ_{VN} this represents number of classes in the public dataset).*

The action recognition accuracy of the different labelling representations can be seen in table 4.5. Over all three datasets, ϕ_{MV} performs comparably with ϕ_{VN} for BEOID (-0.5%), worse on CMU-MMAC (-1.9%) and significantly better on GTEA Gaze+ ($+10.5\%$). The largest improvement can be attributed to the higher overlap in actions in GTEA Gaze+ compared to the other two datasets: The same action, and therefore verbs, are applied to multiple objects and vice versa leading to a much larger number of actions per verb compared to BEOID and CMU-MMAC, as well as significantly more verb-noun classes. ϕ_{SV}, as expected, performs worse than all other approaches due to the ambiguous nature of describing an action with a single verb. However, for GTEA Gaze+ ϕ_{SV} performs only marginally worse than ϕ_{VN}.

Comparing the use of soft or hard assignment leads to ϕ_{MV} performing better on BEOID and CMU-MMAC but getting outperformed on GTEA Gaze+. Generally, ϕ_{MV} seems to perform better on datasets which don't have a large number of overlapping actions.

Conclusion of Action Recognition Results

The results first showed that, for action recognition, using semantic knowledge from WordNet or Word2Vec isn't useful for learning a verb-only representation. Whilst in some cases it can be beneficial for videos where semantic relationships such as synonyms and hyponyms are present, it mostly has a detrimental effect on relating verbs contextually. Next, experiments tested the use of a sigmoid cross entropy loss for learning ϕ_{SAMV} which can be thought of as a regression problem finding that the loss classically used in multi-label learning was able to learn a better model. Finally, for action recognition ϕ_{MV} was found to perform comparatively to a ϕ_{VN} across all three datasets with ϕ_{SAMV} only outperforming on the largest of the three testing datasets, GTEA Gaze+.

4.8.2 Action Retrieval Results

In this section the verb only representations are evaluated on the task of action retrieval.

In order to evaluate $\phi_{SV,MV,SAMV}$ on the action retrieval task, the output of ϕ is treated as an embedding space where each verb is represented by a single dimension. Due to this, singular verbs can be 'embedded' into this space via a one-hot vector whereas videos can be embedded using the values of their label representation. This allows for both video-to-text (VT) retrieval by finding the closest verbs to a video in this space and text-to-video (TV) retrieval by finding the closest videos to a verb[8]. Additionally, the verb labels are also evaluated qualitatively to check their consistency across datasets by providing cross-dataset retrieval results.

Evaluation Metric

For all retrieval tasks, mean average precision (mAP) is used. This finds the mean of the average precision for each query. It is common for recall@k to be used within the literature for retrieval-based tasks, however, mAP is used here to additionally ensure that the returned ranking of the relevant/irrelevant items is evaluated — recall@k only calculates the percentage of correctly retrieved items within the top k. The definition used to calculate mAP is provided below:

To begin, the average precision for a query x_i from a test set modality X is found: The set Y_i^+ is created such that it includes all of the relevant retrievals for x_i. Note that

[8] More verbs can also be used as a query by constructing a binary vector or, if the relevances are to be taken into account, a real vector can be used instead.

$x_i \notin Y_i^+$. Next, the retrieved items are ranked by their distance to the query item x_i, denoted by R_i. The average precision is defined by the precision at m ($Pr@m$) at all places where the mth ranked retrieval is relevant, *i.e.* it is in the set Y_i^+. In summary, the averaged precision of a query is given below:

$$AP(x_i) = \frac{1}{|Y_i^+|} \sum_{m=1}^{|R_i|} Pr@m(R_i) \times I_m \tag{4.10}$$

where $Pr@m(R_i)$ returns the precision of the first m items in R_i and I_m is an indicator function which returns 1 if the mth ranked item is relevant and 0 otherwise.

The mAP can then be calculated by finding the mean of the averaged precision over all queries:

$$mAP(X) = \frac{\sum_i^N AP(x_i)}{N} \tag{4.11}$$

where AP is the average precision calculated in Eq 4.10.

Video-to-Text Retrieval

This section evaluates the ϕ on their ability to retrieve verbs given query video that are correct and (in the case of SAMV) in the correct order.

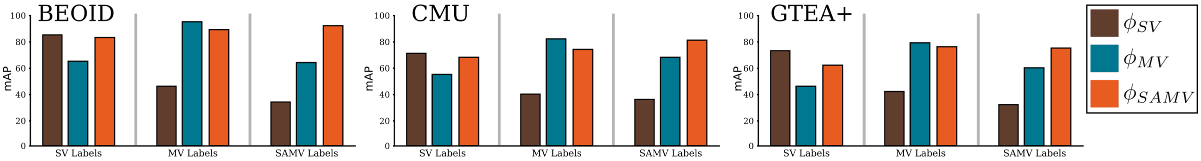


Figure 4.12: *Results of Video-to-Text retrieval across all three datasets. For each ϕ the mAP is shown on each of the three labelling schemes $L \in \{SV, MV, SAMV\}$.*

As each method is simply retrieving verbs, it is possible to perform retrieval of each ϕ on all labelling representations. *I.e.* ϕ_{SAMV} can be used to retrieve the SV ground truth (the majority verb). Figure 4.12 shows the results of these retrievals where for each $\phi_{SV,MV,SAMV}$ the mAP of retrieving the ground truth of each verb-only labelling method ($L \in \{SV, MV, SAMV\}$) is shown. As expected, each ϕ performs best on its own ground truth, however, the generalisability of ϕ_{SAMV} is apparent, showing comparable results to ϕ_{SV} and ϕ_{MV} on the ground truth of both SV and MV. This suggests that a soft assigned multi verb representation is able to learn both the most common verb used to describe an action in addition to the distinction between the set of main verbs and supplementary verbs.

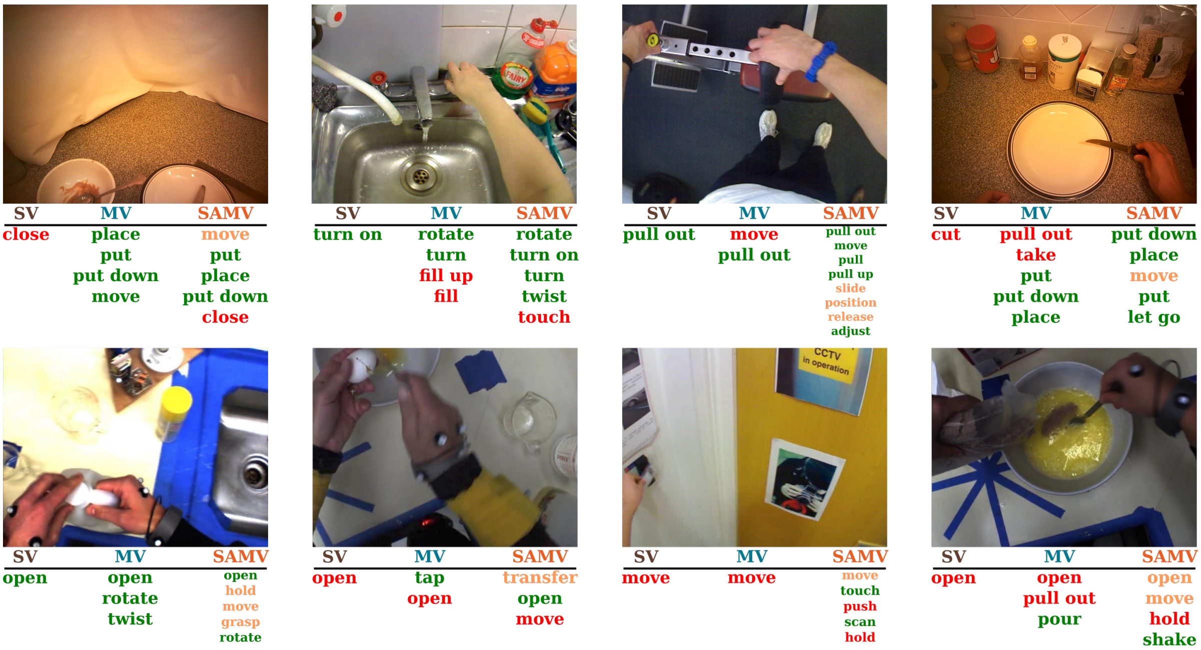


Figure 4.13: *Qualitative results of $\phi_{SV,MV,SAMV}$. Correct and incorrect retrievals are denoted green and red respectively. Orange verbs are those which have been predicted substantially higher than the ground truth ranking. Generally, ϕ_{SAMV} is able to predict more correct verbs than the other two approaches.*

Qualitative results of video-to-text retrieval can be seen in Fig. 4.13 for each verb-only labelling method. Green and red verbs denote correct and incorrect predictions whilst orange verbs represent those which have been retrieved significantly higher than the ground truth ranking. It is noted that ϕ_{SAMV} is able to predict more, correct verbs than the other two representations, though it is common to see *"move"* ranked higher than it should have been.

Text-to-Video Retrieval

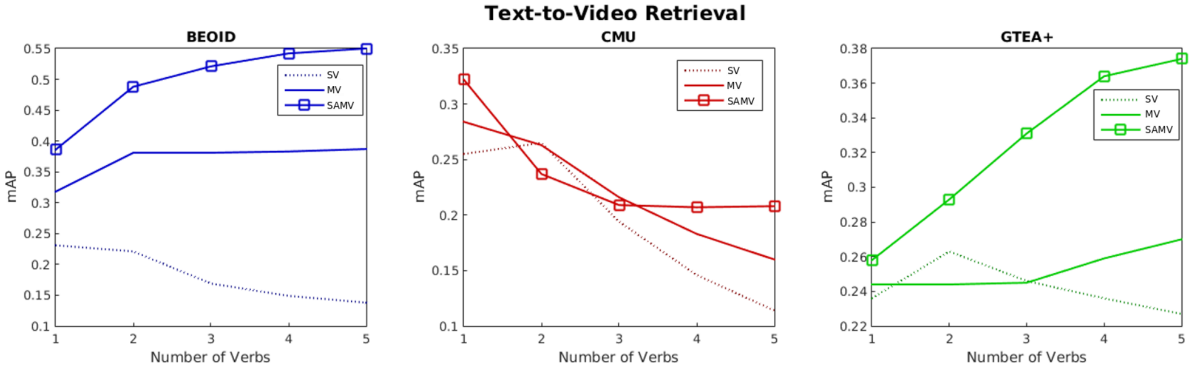


Figure 4.14: *Results of text-to-video retrieval across all three datasets. Queries are tested with an increasing number of verbs and reported using mAP.*

The introduction to this chapter stated how by themselves verbs are ambiguous but using multiple verbs can help overcome this issue. This is evaluated by performing text-to-video retrieval with an increasing number of verbs. The learned embedding space is queried using all combinations of co-occurring verbs from 1 to 5 as binary vectors. The results for $\phi_{SV,MV,SAMV}$ can be seen in Fig. 4.14 for all three datasets. For both BEOID and GTEA Gaze+, the mAP of ϕ_{SAMV} increases as the number of verbs used to query the embedding space increases, causing it to significantly outperform both ϕ_{SV} and ϕ_{MV}. Similarly, for ϕ_{MV}, by increasing the number of verbs used to query, the mAP is either stable or sees an increase suggesting that for main verbs it is important to query with a larger number. For CMU-MMAC there is a drop in performance for all ϕ, which is attributed to the coarser grained nature of the videos (which can be seen in both Fig. 4.7 and when comparing the average lengths of videos: $8.7s$ for CMU-MMAC, $1.6s$ for BEOID and $2.0s$ for GTEA Gaze+). This causes a significantly higher overlap between supplementary verbs, though ϕ_{SAMV} still outperforms alternatives at $n \in [4, 5]$. Given these results it suggests the suitability of ϕ_{SAMV} for the task of text-to-video retrieval.

Cross Dataset Retrieval

Video-to-video (VV) retrieval can be performed across datasets using ϕ_{SAMV} by finding the closest representation of a video in a different dataset. *I.e.* given $\hat{y}_i^{BEOID}$ representing an embedded video from BEOID, what is the closest $\hat{y}_k^L$ where $L \in \{CMU\text{-}MMAC,$

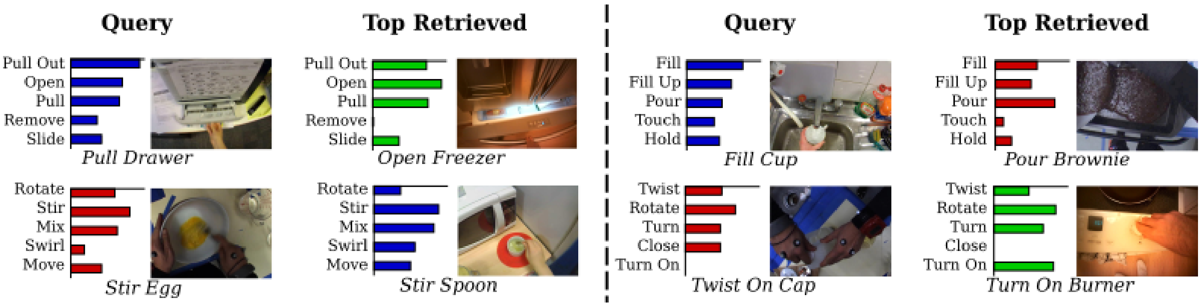
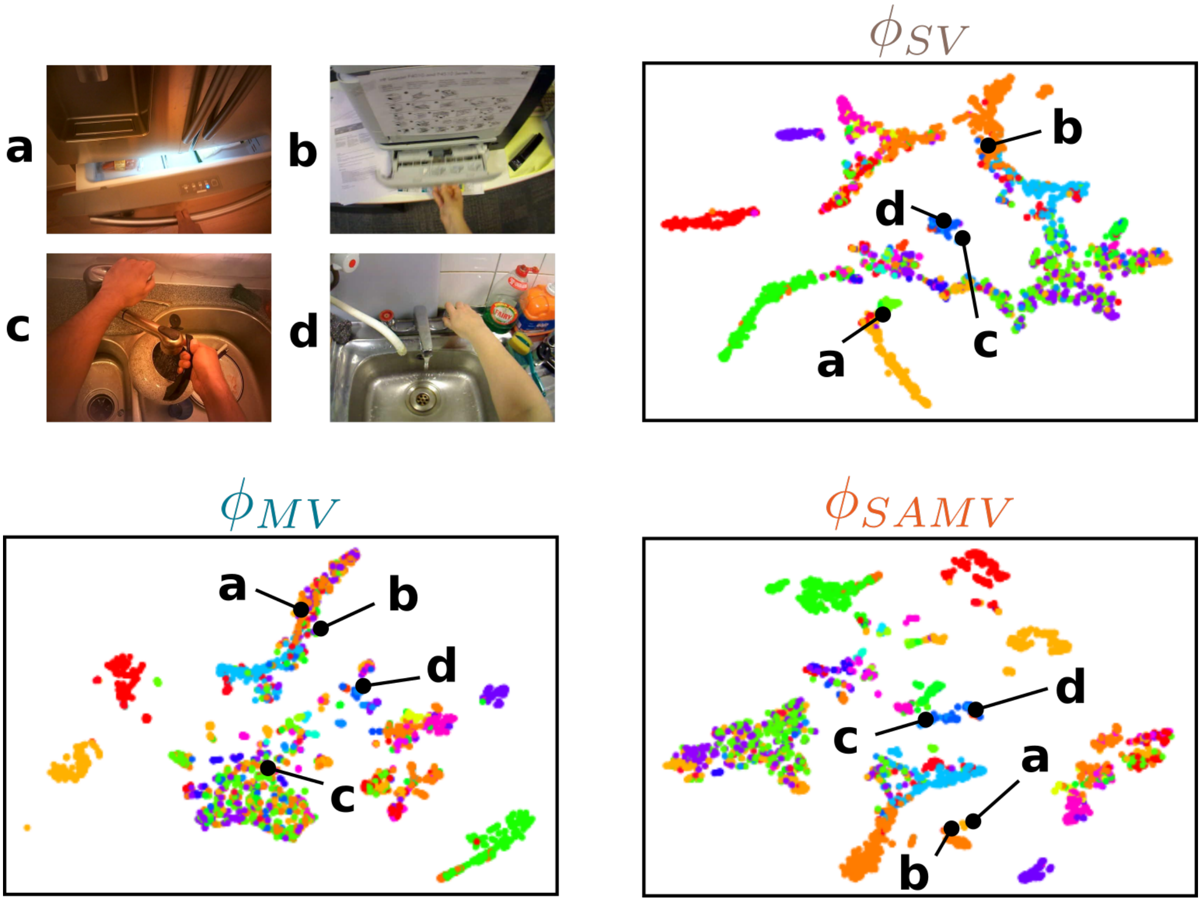


Figure 4.15: *Examples of video-to-video retrieval being performed across datasets using the output of ϕ_{SAMV}.*

GTEA Gaze+}? Figure 4.15 shows qualitative examples of cross dataset retrieval between all three datasets (BEOID in blue, CMU-MMAC in red and GTEA Gaze+ in green).

"Pull drawer" and *"open freezer"* are two actions which describe almost the exact same action, even though the actions are labelled by different verbs and different nouns in the original ground truth labels. Also noteworthy, is ϕ_{SAMV}'s object-agnostic nature in which *"stir egg"* and *"stir spoon"* are found via query. Finally, similar motions but with different end states can be seen as comparable, the query of *"twist on cap"* returns *"turn on burner"*, yet the presence of result verbs allows differentiation between the two.

The embedding space of all three datasets can also be inspected, and this is shown in Fig. 4.16 using a t-SNE representation of videos from all three datasets with $\phi_{\{SV,MV,SAMV\}}$. In each case, the videos are coloured by the verb with the highest soft assignment score. Moving from ϕ_{SV} to ϕ_{MV} to ϕ_{SAMV} the importance of the majority verb is reduced leading to an increased overlap in clusters. Two pairs of examples are highlighted in the left of the figure: *"pull drawer"* and *"open freezer"* along with two examples of *"turn-on tap"*. For the former case ϕ_{SV} places the videos far apart due to the different verbs chosen to label the video (manner vs. result in this case), but ϕ_{MV} and ϕ_{SAMV}, using a multi-verb representation, embed these videos much closer in space. The second example pair includes two taps which are turned off in different ways, pushing down for (c) and rotating for (d). ϕ_{SV} only uses a single verb and is therefore unable to make this distinction. As ϕ_{MV} applies an equal importance score to each verb these videos are actually placed far apart in this embedding, whereas ϕ_{SAMV} embeds these videos close together, but still separable due to the different manners required.


ϕ_{SV}

ϕ_{MV}

ϕ_{SAMV}

Figure 4.16: *T-SNE representations of $\phi_{SV,MV,SAMV}$ with videos from all three datasets. Each video is assigned a colour based on its majority verb. Two pairs of videos are highlighted in this example: "open freezer" and "pull drawer"(a) and (b) as well as "turn off tap" (c) and (d).*

Conclusion of Action Retrieval Results

As expected, using a singular verb as a label leads to a lot of ambiguity causing ϕ_{SV} to perform significantly worse for both video-to-text and text-to-video retrieval. Unlike for action recognition, ϕ_{MV} is outperformed by ϕ_{SAMV} for the action retrieval task with the latter being a suitable substitute for the other three labelling representations for video-to-text retrieval making ϕ_{SAMV} a clear choice for retrieval tasks. This is shown further with the qualitative results of cross dataset retrieval in which ϕ_{SAMV} is able to relate similar motions across datasets regardless of objects being interacted with.

4.8.3 Verb Prediction Error

In this section, the error of verb prediction is evaluated via the root mean square error metric. This section shows how ϕ_{SAMV} is able to not only learn a much larger vocabulary than ϕ_{SV} and ϕ_{MV} but also with a lower error due to the soft assignment of verbs.

Evaluation Metric

The per verb prediction error can be found using the root mean square error (RMSE) per verb using the following equation:

$$E(v_j|\phi) = \frac{1}{|\{x_i; y_{i,j} > 0\}|} \sum_{i;,y_{i,j}>0} ||y_{i,j} - \hat{y}_{i,j}|| \qquad (4.12)$$

Note: this equation only counts the error of verbs that are present in the ground truth of the video as error in assignment score is to be evaluated.

Results

The per-verb RMSE can be seen in Fig 4.17 across all three datasets. Both ϕ_{SV} and ϕ_{MV} struggle to predict the correct values of each verb leading to a large per verb error. ϕ_{SAMV} instead is able to learn a much larger vocabulary of verbs, yet has a much lower verb prediction error.

	BEOID			CMU			GTEA+		
	ϕ_{SV}	ϕ_{MV}	ϕ_{SAMV}	ϕ_{SV}	ϕ_{MV}	ϕ_{SAMV}	ϕ_{SV}	ϕ_{MV}	ϕ_{SAMV}
No. of verbs	20	42	90	12	31	89	15	34	90
RMSE	0.73	0.78	**0.06**	0.78	0.56	**0.09**	0.74	0.67	**0.11**

Table 4.6: *Average RMSE of ϕ_{SL}, ϕ_{ML} and ϕ_{SAMV} for the three different datasets.*

Table 4.6 summarises these results showing the average RMSE per dataset. Generally, increasing the size of the vocabulary causes a reduction in prediction error per verb across all three datasets with ϕ_{SAMV} using the soft assignment scores seeing a significant

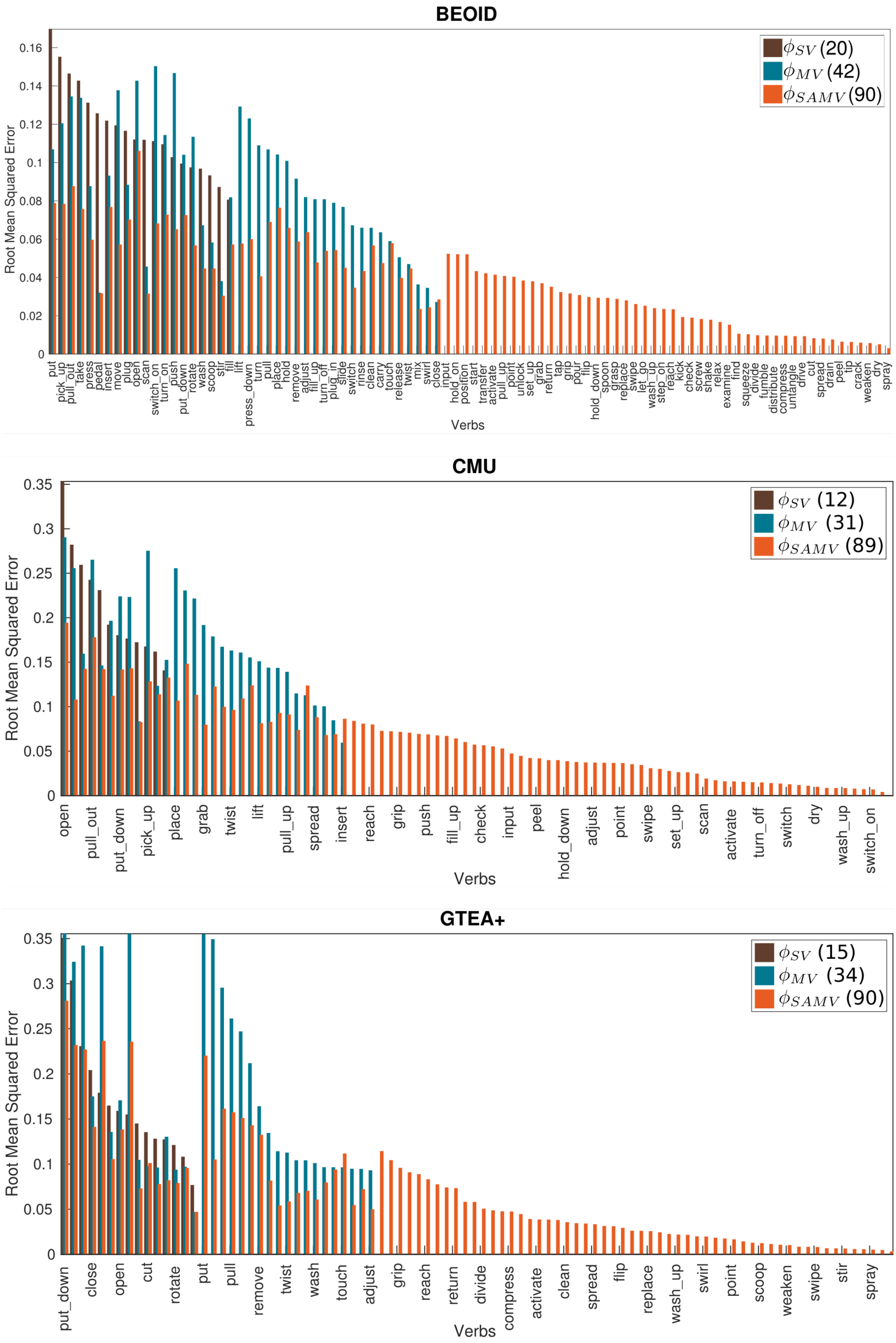


Figure 4.17: *Per Verb RMSE over all datasets for $\phi_{SV,MV,SAMV}$. Verbs are grouped by their presence in the different labelling representations. Number of verbs learned by each representation is shown in the legend.*

decrease in RMSE. Interestingly, ϕ_{SAMV} is able to learn uncommon verbs very well, with *"crack"* and *"spray"* having a very low error even though they are very uncommon in the annotations (see figure 4.7).

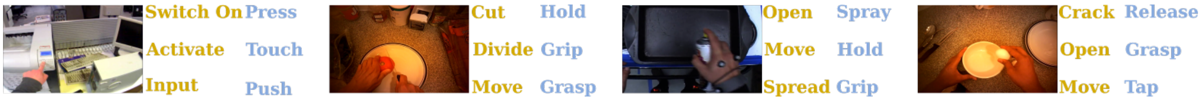


Figure 4.18: *Qualitative video-to-text retrieval results of manner (blue) and result (yellow) verbs showing examples of their relationship.*

Figure 4.18 includes qualitative retrievals for result and manner verbs using ϕ_{SAMV} showing the top 3 result (yellow) and manner (blue) verbs for four different actions. Quantitatively, ϕ_{SAMV} learns both manner and result verbs similarly across all three datasets, seen in table 4.7.

RMSE	BEOID	CMU-MMAC	GTEA Gaze+
Manner	0.051	0.094	0.107
Result	0.061	0.077	0.102

Table 4.7: *Average RMSE of ϕ_{SAMV} for manner verbs and results verbs over all datasets.*

4.9 Conclusion

In conclusion, this chapter has presented the notion of using verb-only labels for action recognition through the use of multiple verbs, overcoming the issue of using a singular verb to label a video. It has highlighted the different types of verbs that are present in a multi-verb representation as well as the relationships between verbs. Notably, most verbs that can be used to describe a video aren't related semantically, rather they are related through context given by the action.

Due to this issue, semantic knowledge bases, such as WordNet and Word2Vec couldn't be used solely to discover these multi-verb representations and accordingly annotations for three datasets (BEOID, CMU-MMAC and GTEA Gaze+) were collected. Two multi-verb representations were proposed along with the standard verb-noun representation and the single verb representation from chapter 3 to be compared against for the task of action recognition.

The results show the benefits of a multi-verb representation for action recognition, with the soft assigned representation being useful for the largest dataset which includes a large overlap of actions. The multi-verb representations could also thought to be an embedding space and so the task of action retrieval could be evaluated when comparing the different verb-only labelling representations. For this task, the soft assigned multi-verb representation was also found to outperform the other representations for the task of video-to-text and text-to-video retrieval.

The multiple verb representations, as discussed in this chapter, could only be collected through the use of multiple annotators for each example video. This represents a large annotation effort in comparison to the single-verb or verb-noun labelling representation. Accordingly, the next chapter will look into how this drawback can be eased, whilst still keeping the generalisability and performance of the soft-assigned approach.

Chapter 5

Part of Speech Embeddings for Large-Scale Datasets

Chapter 4 introduced multi-verb, verb-only representations for action recognition and action retrieval. Results showed that ranking verbs (*i.e.* by using a soft assignment score per verb) outperformed the other verb-only representations for action retrieval. However, the annotation effort for even the largest dataset, GTEA Gaze+, was considerable. With the popularity of deep learning, large-scale datasets are becoming more and more common in order for successful training, leading to the multi-verb approach being unscalable.

This chapter explores this issue two-fold: Firstly, it presents an exploration into categorising verbs and nouns for a large-scale dataset, EPIC-Kitchens, which used an open vocabulary during collection. Secondly, it presents a method which embeds actions into a unified space creating separate part of speech embeddings, building upon the findings of the multi-verb representations of the previous chapter for the task of fine-grained action retrieval.

As EPIC-Kitchens was collected with an open vocabulary via narration, the original annotations are quite different to the standard verb-noun class labels that have been used for action recognition. Whilst both a verb and a noun are present in the annotation, many contained additional nouns or other parts of speech (such as adjectives or adverbs) leading to each video being labelled with a short caption. These captions provide another route for verb understanding as well as understanding of other parts of speech which was the motivation for the proposed method which disentangles the video captions into their constituent parts of speech.

The previous chapter introduced the task of action retrieval: Given a video can the relevant verbs be retrieved. However, with the short form captions, in addition to the disentanglement of other parts of speech, other forms of retrieval can be evaluated. Namely, retrieving the relevant nouns and retrieving the relevant captions. This chapter also presents a formulation of the latter retrieval task, named as fine-grained action retrieval, noting its differences to the general video retrieval task introduced in section 2.3.2 Finally, the notion of disentangling different parts of speech is also applied to the general video retrieval task, noting its benefits, and a full study on how different parts of speech change the underlying embedding.

This chapter is broken down as follows: Section 5.1 details the EPIC-Kitchens dataset providing information on how it was collected with an open vocabulary and how the closed vocabulary classes were created. Section 5.2 discusses how the closed vocabulary classes of EPIC-Kitchens, in addition to the original open vocabulary captions, can be used for retrieval and, specifically, on the fine-grained action retrieval task. The proposed method is given in section 5.3 before experiments on EPIC-Kitchens and MSR-VTT [144] are conducted in sections 5.4 and 5.5 respectively.

5.1 An Open Vocabulary Action Recognition Dataset

EPIC-Kitchens [23] was created to be a large-scale dataset in the field of egocentric vision. Additionally, the dataset was collected with two aims relevant to this thesis: Firstly, scaling the creation of a large dataset and, secondly, with the aim of being as realistic as possible. This section includes a brief overview of the collection process of EPIC-Kitchens in addition to its make-up before discussing the pertinent parts of the process to this thesis, namely being the method in which the open vocabulary of the EPIC-Kitchens dataset can be clustered for use in action recognition.

5.1.1 Dataset Collection

EPIC-Kitchens was collected by 32 different participants in 32 unique kitchen environments across 4 different countries. Each participant was asked to start the video recording from the moment they stepped foot inside their own kitchen to when they left it[1].

After recording for 2-3 days, participants were then asked to narrate their clips. They were asked to provide short sentences including a single verb with one or more nouns to describe the fine-grained actions they completed. It was imperative that the participants recorded their own videos in order to make sure the action description was correct. This narration was collected by the participants in real-time watching back the videos they had recorded. They had a choice of their native language or English depending on which they felt more comfortable with.

The narrations were then transcribed and, where applicable, translated to English, both via the use of Amazon Mechanical Turk (AMT). Once the action had been transcribed, AMT workers could then choose suitable start and end times for an action segment. Each action would therefore consist of a short phrase consisting of a verb and one or more nouns and an accompanying video segment (length: $\mu = 3.7s$, $\sigma = 5.6s$). Note that while multiple language captions exist in EPIC, only English captions were used within this thesis.

5.1.2 Dataset Make-Up

As a whole, EPIC-Kitchens contains 39.6K action segments over training and testing. In order for the generalisability of methods to be tested on EPIC-Kitchens, it was decided to create two different test sets based on the kitchens that the dataset was collected in. In the training set only 28 of the 32 kitchens are present allowing for the creation of two test sets. The first contains test examples from the 28 kitchens in the training set, labelled as the SEEN test set, and the second which contains all of the action segments from the 4 withheld kitchens, called the UNSEEN test set. In this way, the UNSEEN test set can be used to show the generalisability of a method on EPIC via the use of

[1] The kitchen as a home environment sees humans performing a large variety of different fine-grained actions compared to the living room or other areas in the home.

the unseen environments. Additionally, due to the open vocabulary used during the collection process, and splitting of the dataset into train and test, it is common for the test sets to contain classes which were not seen in testing, *i.e.* zero-shot classes. This is explored in more detail in chapter 6. Overall, the training set contained $28,561$ action segments, the SEEN test set $8,064$ action segments and the UNSEEN test sets $2,939$ action segments.

5.1.3 The Open Vocabulary of Epic

As the annotations of EPIC-Kitchens were collected with participants narrating over the videos that they had recorded, they were not limited in any way as to how they described the actions. In fact, the participants were encouraged to use words and phrases which felt natural to them, *i.e.* not force them to call a *"baking tray"* a *"pan"* or use the verb *"place"* instead of *"put"* as well as translations diversifying the vocabulary even further. The captions were parsed using a combination of models from SpaCy [3][2]. Because of this, the 39.6K action segments included over 629 different verbs and $1,240$ different nouns. Table 5.1 shows the counts of some example verbs and nouns from all parts of the long-tail distribution.

For the action recognition task, this list of verbs and nouns would prove impossible to train a standard one-*vs.*-all classifier for. Both the list of verbs and nouns contained a large number of overlaps between different words. This would lead to many of the problems discussed in chapters 3 and 4. Notably, that using an open vocabulary cannot be solved using hard assignment of singular labels.

In order for the labels to be usable for action recognition, which classically requires a closed vocabulary and a single-label setup, the verbs and nouns were clustered. Each cluster was constructed such that all of the words within a single cluster would be interchangeable for the purposes of describing the action in addition to having minimal overlap with other clusters. In this sense, if the video contained someone putting a chopping board on the counter, then *"place"* or *"put"* would be a valid candidate for a verb but *"move"* would not. Similarly, *"cutting board"* or *"board"* would be a valid noun and *"surface"* would not be. This means that the verb clusters contain main verbs (see figure 4.6) which are all synonymous with each other whereas all of the nouns

[2] The small, medium and large English parsing models were used in a cascade fashion, if the small model found no verbs then the medium model was employed *etc.*

Verb	Count	Noun	Count
put	4310	tap	1531
open	3734	plate	1263
pick-up	3520	knife	1184
take	3175	drawer	984
put-down	2931	spoon	968
close	2492	pan	947
rinse	1675	cupboard	925
wash	1619	fridge	836
stir	934	hand	812
pour	890	bowl	798
place	774	lid	785
get	730	glass	654
cut	708	onion	646
move	653	fork	597
turn-on	576	water	546
...	...	...	...
start	24	salmon	45
knead	23	tuna	43
put-back	23	measuring cup	42
take-off	23	temperature	41
push	22	avocado	41
fill-up	22	packet	41
pour-up	22	utensil	41
change	21	lime	40
roll	20	v60	40
toss	20	wrap	39
transfer	18	strainer	39
rinse-off	17	grill	39
place-in	17	pesto	38
take-up	16	tin	38
spray	16	clothes	37
...	...	...	...
knife	1	stove:gas	1
open-to	1	laundry liquid	1
regulate	1	plastic spoon	1
take-into	1	coconut jar	1
plug	1	pizza pan	1
remove-inside	1	coke	1
drape	1	cake container	1
get-in	1	inside microwave	1
tip-in	1	plain flour	1
turn-muffin	1	electronic kettle	1
unroll	1	turkey breast	1
reseal	1	cooking time	1
squish-into	1	broccoli sort	1
plate-on	1	cheese container	1
twist	1	other glass	1

Table 5.1: *Examples of the open vocabulary of EPIC for both verbs and nouns.*

within a cluster are valid synonyms or contextually relevant within EPIC-Kitchens. The non-overlapping clusters would allow for EPIC-Kitchens to be trained for the standard single-label classification task.

5.1.4 Clustering an Open Vocabulary

From chapters 3 and 4 the verb hierarchy for WordNet has been found to be poor in comparison to the noun hierarchy, which is well known for its ability to relate nouns within the literature. Because of this, a manual approach was adopted for the verb clustering and WordNet along with the Simplified Lesk algorithm [62, 71] was attempted for noun disambiguation and clustering.

Simplified Lesk Algorithm The simplified Lesk Algorithm is used for the task of word sense disambiguation. Specifically, given a dictionary of words and their different senses (*e.g.* meanings or synsets in WordNet) in addition to a sentence, the simplified lesk algorithm tries to disambiguate the meaning of the word by computing the overlap between the sentence, excluding stop words, and the definitions of the different word senses.

For example, the word *"knife"* has three different synsets in WordNet with the following definitions:

- **knife.n.01** *"Edge tool used as a cutting instrument; has a pointed blade with a sharp edge and a handle."*
- **knife.n.02** *"A weapon with a handle and blade with a sharp point."*
- **tongue.n.03** *"Any long thin projection that is transient."*

Given a sentence from EPIC-Kitchens, such as *"cut tomato with knife"* knife.n.01 would be given as the word sense due to the overlap between *"cut"* in the caption and *"cut[ting]"* in the definition.

Immediately a drawback of using the simplified lesk algorithm on the captions from EPIC-Kitchens can be seen in that the captions themselves are very short[3]. With most

[3] Other word sense disambiguation methods can improve upon the results of the simplified lesk algorithm, but they require a longer caption, or indeed at least a full sentence still performing badly on EPIC-Kitchens.

captions only consisting of a verb and a noun[4] there is very little overlap between the definitions and the captions leading to most of the nouns within EPIC-Kitchens being given an incorrect synset. *E.g.* for the caption *"pick up knife"* there is no overlap with any of the three definitions listed above. In this case, the most common synset is often chosen but frequently this can cause erroneous word sense disambiguation. Because of the number of errors of using the Simplified Lesk Algorithm, and the manual checking that was required, the notion of automatically clustering the nouns was dropped and a semi-manual clustering approach was used instead, the details of which can be found next.

Manual Clustering For the verbs, a full manual clustering was performed. The entire list of 629 verbs was collated and then 6 researchers, who provided recordings for the dataset, went through and grouped verbs which were deemed to be interchangeable regardless of context. Multiple rounds of revisions took place by the participants until no major changes were made to the clustering, and an agreement had been met.

An automatic first pass was used for the nouns where similar nouns were bagged together first. This was done by splitting the compound nouns with the final noun in the pair being chosen as a cluster. If the noun constituted of a single word then that was used in a cluster. This means that nouns such as *"baking pan"*, *"pan"* and *"large pan"* would be automatically clustered with *"pan"* whereas *"baking tray"* or *"saucepan"* would not: baking tray would be put with other trays and saucepan is a single word. The nouns then went through the same process as the verbs with 6 participants going through and manually verifying and updating the clusters.

The result of this clustering meant that the 629 verbs and $1,240$ nouns were reduced to 125 verb clusters (average of 3.50 ± 3.91 verbs per cluster) and 331 noun clusters (average of 2.69 ± 3.64 nouns per cluster). For example, the *"close"* cluster includes the verbs *"close"*, *"close-off"* and *"shut"*, whereas the noun cluster *"cheese"* includes *"cheese slice"*, *"mozzarella"*, *"paneer"*, *etc.*

[4] Mean length of captions in the EPIC-Kitchens training set is 2.85 ± 1.62.

5.1.5 Defining Actions Classes from Clusters

With the individual verb and noun clusters found, each video still required an action class for the purposes of action recognition. The action class was defined as a cross product between the verb class and the *first* noun class assigned to each video, leading to the verb-noun action representation introduced in section 4.4. The extension to multi-label noun recognition was decided to be left for future work.

This gave a total of $3,033$ different action classes with an average of 13.0 examples per class. As with the verbs and nouns, this set heavily follows a long tail distribution with 36.5% of actions only containing a single example.

Many Shot and Zero Shot The construction of the dataset naturally lends itself to many shot, few shot and indeed zero shot tasks. A many shot training set was created by choosing examples of which there were > 50 examples per verb and noun class within the training set. This creates a list of $1,836$ different action classes from the 26 different many shot verbs and 314 different many shot nouns and a reduction of $1,762$ videos or 6.19%. This has the benefit of a much larger number of zero shot verb and noun classes present in the SEEN and UNSEEN test sets, see chapter 6 for more details.

5.2 From Classes to Captions

Previous chapters have been focused, at least in part, on the task of (video) action recognition. That is, given a video return the correct class. This chapter focuses solely on retrieval in which, given a query item, the task is to retrieve all relevant items. For this task there is not a reliance on classes and instead there is a move towards captions being used as class labels. This change in focus was motivated due to the benefits shown at the end of the previous chapter in verb understanding and the ability to learn an open vocabulary.

The focus shifts from classes to captions. Simply, these captions could be the verb-noun action class as was defined in section 4.4 that was created for EPIC-Kitchens on the action recognition task, but it is often that datasets used for the general video retrieval task contain much longer captions. Generally, these captions are loosely structured as sentences which can range from as small as a couple of words to as long as twenty words.

As mentioned in the literature review for general video retrieval (section 2.3.2), datasets are constructed with the idea of matching the cross-modal pairs together, via instance-based relevancy. This is somewhat limiting as two captions which are similar but belong to different videos are classed as irrelevant. The clusters that were created for EPIC-Kitchens in section 5.1.4 can be used as a way of overcoming this issue and provide semantic relevancy information for not only the standard cross-modal retrieval tasks (*i.e.* video-to-text and text-to-video retrieval) but also within-modal retrieval tasks (*i.e.* video-to-video and text-to-text retrieval). For example, the *"put"* cluster allows for *"put down"*, *"place"* and all other verbs within the cluster to be treated as valid retrievals for the fine-grained action retrieval task. Nouns can be treated similarly, giving a similar ability to know the relevant and irrelevant retrievals for each noun. The combination of verb and noun, as with the construction of the action classes, naturally create valid and invalid action retrievals.

However, even with only verbs and nouns the size of possible actions can be very large. Using EPIC-Kitchens as an example, which has 125 verb classes and 352 noun classes, there are a maximum number of $44,000$ different action classes. Of course, a lot of these are nonsensical verb-noun pairs (*e.g.* pour oven) with only $3,151$ combinations occurring in either the training or test sets. This raises another important issue with using EPIC-Kitchens and other open vocabulary datasets: The long tailed nature of the actions. Of the $3,151$ different actions only 568 have more than 10 occurrences and 149 of those have more than 50 occurrences. Given the small number of instances, this chapter proposes learning these actions via disentangling the caption into its constituent parts of speech. This has three main benefits: Firstly, this allows for better understanding of the individual parts of speech. Secondly, the part of speech tags can be found with no additional training effort by using a part of speech tagger and allows the injection of this additional information. Finally, by considering only a single part of speech in turn, *i.e.* verbs, it allows for generalisation as it forces learning of the verbs in different contexts without regard to the other parts of speech. In this way while there might be a small number of instances of *"put meat"* by disentangling the caption the method can learn *"put"* and *"meat"* separately for different contexts.

Retrieval on EPIC-Kitchens therefore consists of a hybridised problem, lying in between fine-grained action recognition and the general video retrieval task that is common in the literature. It differs from the latter[5] in three main aspects: Firstly, relevance is defined semantically (using the verb/noun clusters from EPIC-Kitchens) as opposed to the instance based approach. Secondly, captions are generally shorter as the actions they describe are shorter. Finally, the long-tailed distribution of the action frequency, due to the fine-grained nature, leads to some actions being orders of magnitude more common than others.

This task, presented under the name of *fine-grained action retrieval*, attempts to perform retrieval on a cross-modal pair that consists of a short video containing a fine-grained action and a small caption.

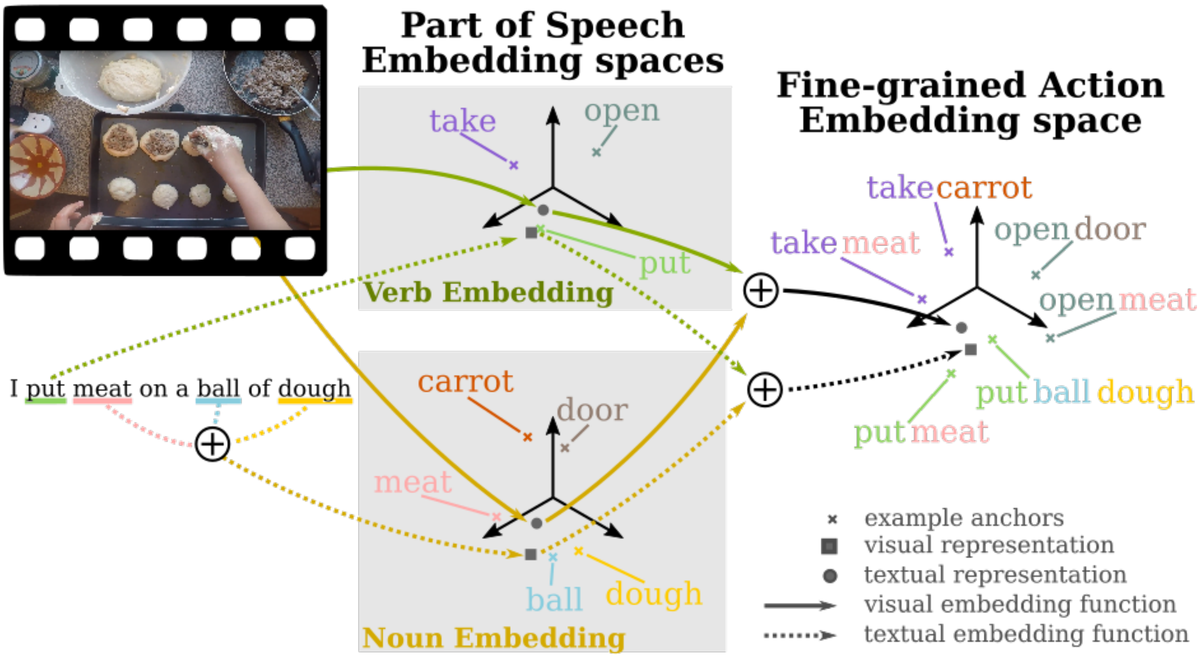


Figure 5.1: *Large-Scale action recognition datasets are often long-tailed in nature: They include a large number of actions with the majority of videos representing a minority of action classes. Splitting an action into its constituent parts of speech allows for direct learning of these minority classes whilst also allowing generalisation of the part of speech embeddings.*

An example of the employed approach can be seen in Fig. 5.1 in which the video has the caption *"I put meat on a ball of dough"*. This is separated into verbs (*"put"*) and nouns (*"meat, ball, dough"*) and two different embeddings are created, one for each part of speech, in which the video and the caption are both embedded. The results of these individual part of speech embeddings are combined into an action embedding. This

[5] See section 2.3.2 for more details of the this task.

final embedding space (in addition to the part of speech embedding spaces) allow for retrieval of both videos and captions given a query item of either. Essentially, all four possible pairs of retrieval can be performed: video-to-video, video-to-text, text-to-video and text-to-text retrieval to retrieve relevant verbs, nouns or actions.

It is noted here that in an action recognition setting the video from Fig. 5.1 would only have the label of *"put meat"* (see section 5.1.5) and so the extra information about the action taking place would be lost. Captions increase not only the number of words but the number of parts of speech as the descriptions become sentences: From purely verbs and nouns in action classes, captions include other parts of speech in the form of pronouns, adjectives, adverbs, *etc.* providing more knowledge of the contents of the video whilst also increasing the space of actions considerably. This also allows exploration into the different parts of speech that make up datasets as well as discovering how useful the different parts of speech are for different datasets, which can be seen in section 5.5.

5.3 Joint Part of Speech Embeddings, (JPoSE)

This section introduces the method in which an action embedding can be learned via the use of multiple part of speech embeddings. First, a multi-modal embedding is defined (section 5.3.1), before describing the training process of a part of speech embedding (section 5.3.2) and finally the entire system in which the part of speech embeddings are learned jointly with the action embedding is detailed (section 5.3.3). An overview of the method can be seen in figure 5.2.

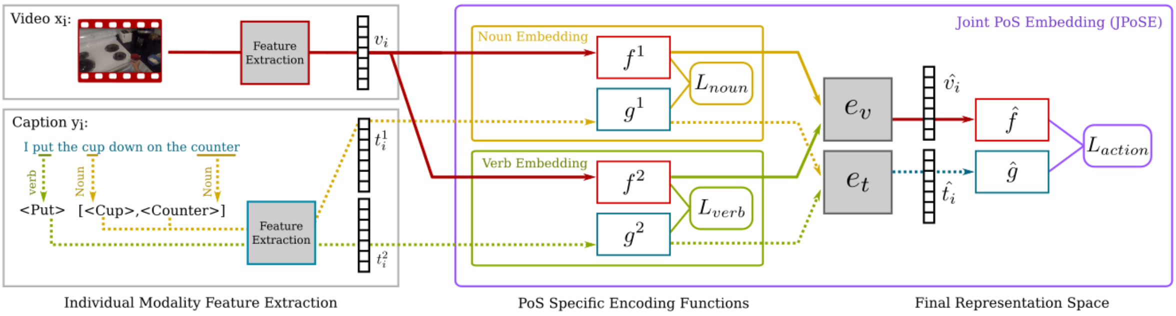


Figure 5.2: *Overview of the Joint Part-of-Speech Embeddings (JPoSE) method. A caption is disentangled into its constituent parts-of-speech and separate multi-modal embedding networks are learned for each part-of-speech (in this example verbs and nouns). The output of the part-of-speech embeddings are combined and passed through a final embedding to learn an action embedding. The visual and textual embedding functions are respectively denoted by f and g with $L_{\{verb,noun\}}$ representing the part-of-speech embedding losses and L_{action} the action loss. Non-trained modules,* i.e. *modules with frozen weights or pooling operations, are shown in grey.*

5.3.1 Multi-Modal Embeddings

This section describes a Multi-Modal Embedding Network (MMEN) which can be used to embed two items from different modalities (in this example videos and captions though others can be used) in the same space.

Let $\{(v_i, t_i)|v_i \in V, t_i \in T\}$ be a multi-modal object, with v_i representing the *ith* video from the (totally ordered) set of videos V, and t_i representing its corresponding caption (*i.e.* the *ith* caption) from the (totally ordered) set of captions T. The aim is to learn two embedding functions $f : V \to \Omega$ and $g : T \to \Omega$, such that $f(v_i)$ and $g(t_i)$ are close in the embedded space Ω[6]. Note that f and g can be simple linear projection matrices or, in the case of this chapter, deep neural networks — the weights of which are given by θ_f for f and θ_g for g. These functions are learned jointly with a weighted combination of cross-modal and within-modal triplet losses (described below). Note that other losses which list, compare pairs or are point-wise losses are valid alternatives to the triplet ranking loss so long as the objective that relevant items are close together in the embedding space and irrelevant items are far apart is held.

[6] Additionally, semantically relevant videos and captions should also lie close to $f(v_i)$ and $g(t_i)$.

Cross-Modal Losses These losses are central to the task of embedding multi-modal objects near each other in the output space. *I.e.* They ensure that the embedded representations of the query item and a relevant item for that query are closer than the representation of the query item with that of a non-relevant item.

Firstly, sets of triplets, $\mathcal{T}$, are created for each pair of cross modalities:

$$\mathcal{T}_{v,t} = \{(i, j, k) \,|\, v_i \in V, t_j \in T_{i+}, t_k \in T_{i-}\} \tag{5.1}$$

$$\mathcal{T}_{t,v} = \{(i, j, k) \,|\, t_i \in T, v_j \in V_{i+}, v_k \in V_{i-}\} \tag{5.2}$$

where T_{i+}, T_{i-} respectively define sets of relevant and non-relevant captions for the *ith* caption and V_{i+}, V_{i-} define the same set of relevant and non-relevant videos for the *ith* video, note that intrinsically v_i is always relevant to t_i for all i. These sets of triplets are then used in the cross-modal triplet losses [137]. Note for brevity f_{v_i} represents $f(v_i) \in \Omega$ and similarly g_{t_i} denotes $g(t_i) \in \Omega$.

$$L_{v,t}(\theta) = \sum_{(i,j,k)\in\mathcal{T}_{v,t}} max\big(m + d(f_{v_i}, g_{t_j}) - d(f_{v_i}, g_{t_k}), 0\big) \tag{5.3}$$

$$L_{t,v}(\theta) = \sum_{(i,j,k)\in\mathcal{T}_{t,v}} max\big(m + d(g_{t_i}, f_{v_j}) - d(g_{t_i}, f_{v_k}), 0\big) \tag{5.4}$$

where m is a constant used as the margin, $\theta = [\theta_f, \theta_g]$, and $d(.,.)$ is a distance function in the embedded space Ω. Here v, t describes the video-to-text loss and t, v the text-to-video loss.

Within-Modal Losses These secondary losses, also called structure preserving losses [135, 137], ensure that the neighbourhood structure within each modality is preserved in the learned space. Intuitively, this helps ensure that by learning the cross modal space items from the same modality have their distances somewhat preserved. Similarly to the cross-modal losses, the sets of triplets are first created for both the video-to-video and text-to-text modalities:

$$\mathcal{T}_{v,v} = \{(i, j, k) \,|\, v_i \in V, v_j \in V_{i+}, t_k \in V_{i-}\} \tag{5.5}$$

$$\mathcal{T}_{t,v} = \{(i, j, k) \,|\, t_i \in T, v_j \in V_{i+}, v_k \in V_{i-}\} \tag{5.6}$$

using the same notation as the cross-modal losses, the within modal losses can then be defined as:

$$L_{v,v}(\theta) = \sum_{(i,j,k)\in\mathcal{T}_{v,v}} max\big(m + d(f_{v_i}, f_{v_j}) - d(f_{v_i}, f_{v_k}), 0\big) \tag{5.7}$$

$$L_{t,t}(\theta) = \sum_{(i,j,k)\in\mathcal{T}_{t,t}} max\big(m + d(g_{t_i}, g_{t_j}) - d(g_{t_i}, g_{t_k}), 0\big) \tag{5.8}$$

MMEN Loss The final loss for training the parameters $\theta = [\theta_f, \theta_g]$ of the multi-modal embedding network is a weighted combination of all four losses described above using all triplets in $\mathcal{T}$:

$$L(\theta) = \lambda_{v,v}L_{v,v} + \lambda_{v,t} + L_{v,t} + \lambda_{t,v}L_{t,v} + \lambda_{t,t}L_{t,t} \tag{5.9}$$

where $\lambda_{\{vv,vt,tv,tt\}}$ are the weights for each loss term.

5.3.2 Part of Speech Embeddings

The last section introduced a multi-modal embedding network, generalised for any usage. This section describes the disentangled part of speech embeddings that are used in the Joint Part of Speech Embedding (in Fig. 5.2 these are the noun and verb embeddings).

To create the part of speech multi-modal embedding network (PoS-MMEN), the caption is first broken down into its constituent parts of speech, grouping words by their tags. For example, *"I divided the onion into pieces using wooden spoon"* can be divided into verbs, [*divide, using*], pronouns, [*I*], nouns, [*onion, pieces, spoon*] and adjectives, [*wooden*]. Part of Speech Embeddings could be created for each of these types, however, as the captions of the EPIC-Kitchens dataset contain mostly verbs and nouns, in addition to clusters only being available for these parts of speech, only verb and noun embeddings are created[7]. Each embedding is trained using the MMEN as described in the previous section, expanded on below, creating a space where (groups of) words with the same part of speech tag lie close to videos in which they describe the video in part.

In order to train the PoS-MMENs, the method of relevancy is adapted from the standard multi-modal Embedding Network, which impacts the make-up of the sets, $V_{i+}, V_{i-}, T_{i+}, T_{i-}$. Specifically, the notion of relevance for the set V_{i+} includes all videos which are related to the PoS portion of v_i. For example, the caption *"cut tomato"* is split into verbs, *"cut"*, and nouns, *"tomato"*. Considering a verb PoS-MMEN the caption *"cut carrot"* would be relevant and a part of T_{i+} due to the shared verb between them whereas the caption *"place tomato"* would be irrelevant and exist in the set T_{i-}. For a noun PoS-MMEN the relevancy of the captions flips for this example, with *"cut carrot"* becoming irrelevant to *"cut tomato"*, whereas *"place tomato"* is now a relevant caption.

The same final loss to train the MMEN, equation 5.9, is used to train the PoS-MMENs, though the loss is given the name of PoS-aware as they focus on the individual parts of speech.

It is important to note that, whilst the inputs to the text embedding functions, g, differs for different PoS-MMENs, the visual input stays the same. In this sense, each visual embedding function, f, should learn different projections and can be seen as learning multiple views of the video sequence for the relevant part of speech.

These PoS-MMENs can be used for part of speech specific retrieval tasks, either verb retrieval for the verb PoS-MMEN, retrieving the correct verb/action in the video/caption, or noun retrieval, retrieving the correct noun/object in the video/caption. Results of these part of speech retrieval tasks can be found in section 5.4.3.

5.3.3 Joint Embeddings

This section now describes how the outputs of individual Part-of-Speech Multi-Modal Embedding Networks can be combined to create an action embedding that is able to be used for action retrieval.

Firstly, the embedding functions of the *kth* PoS-MMEN can be denoted by $f^k : V \rightarrow \Omega^k$

[7] The JPoSE method is generalisable to include more parts of speech, see section 5.3.3

and $g^k : T \rightarrow \Omega^k$ for the visual and textual embedding functions with parameters $\theta^k = [\theta^k_f, \theta^k_g]$ respectively and Ω_k representing the output space of the kth PoS-MMEN.

With the objective of learning an output space that combines the output spaces of the K different Part of Speech Multi-Modal Embeddings, the embedded visual items and embedded text items can be given as:

$$\begin{aligned} \hat{v}_i &= e_v(f^1_{v_i}, f^2_{v_i}, ..., f^K_{v_i}) \\ \hat{t}_i &= e_t(g^1_{t_i}, g^2_{t_i}, ..., g^K_{t_i}) \end{aligned} \tag{5.10}$$

where e_v and e_t represent encoding functions that combine the outputs of the K PoS-MMENs into a space of visual representations, $\hat{V}$, and textual representations, $\hat{T}$. Different forms for the encoding functions are tested using pooling operations, *concatenation, max, average*[8].

Now, considering the combined Part of speech representations $\hat{v}_i$ and $\hat{t}_i$ as input, more advanced functions can be used which further embed $\hat{v}_i$ and $\hat{t}_i$ into an action embedding. Formally, the parameters $\hat{\theta}_{\hat{f}}$ and $\hat{\theta}_{\hat{g}}$ of two embedding functions $\hat{f} : \hat{V} \rightarrow \Gamma$ and $\hat{g} : \hat{T} \rightarrow \Gamma$ will transform the combined Part of Speech representations into an action embedding. This is done by considering the task as another Multi-Modal Embedding Network with the inputs of $\hat{v}_i$ and $\hat{t}_i$, following the process in section 5.3.1. However, the notion of relevance differs from that described in the previous section and so the triplets present in $\mathcal{T}_{v,t}, \mathcal{T}_{t,v}, \mathcal{T}_{v,v}, \mathcal{T}_{t,t}$ are thus modified for action relevancy. That is, a video/caption is relevant to another video/caption if both the verbs and the nouns are the same[9]. For example, the caption *"I cut tomato"* would only be relevant to another caption with a verb synonymous to *"cut"* and a noun synonymous to *"tomato" e.g. "I slice tomato"*. In the case where multiple nouns are present the first noun is used to determine the relevancy. So *"I cut tomato using knife"*, *"I cut tomato using fork"* and *"I cut tomato"* are all considered relevant whereas *"I cut carrot using knife"* would be considered irrelevant. This is done using the assumption that the first noun within the caption is the most important, and thus is more key to differentiating the action. In the case above the knife is simply the tool for the action: It could be omitted and the caption would still make sense, the reverse is not true. These triplet sets are thus used

[8] The latter two of which require that Ω_k has the same dimensionality for all k

[9] In the generalised case, it might be expected that this forces for each Part of Speech the words to be identical in order for the actions to be relevant. However, this may not be useful depending on the Parts of Speech present. For example, the caption *"I beat the eggs rapidly"* and *"he whisks the eggs thoroughly"* could be argued to represent a relevant pair even though the pronouns and adverbs are different. This exploration is left for future work.

with the loss in equation 5.9 which is denoted as $\hat{L}(\hat{\theta})$.

The system can be trained jointly, using the losses for each of the underlying Part of Speech Multi-Modal Embedding Networks, in addition to the action loss as described below:

$$L(\hat{\theta}, \theta^1, ..., \theta^K) = \hat{L}(\hat{\theta}) + \sum_{k=1}^{K} \alpha^k L^k(\theta^k) \tag{5.11}$$

where α^k are weighting factors for the Part of Speech Embeddings. This loss is used to train the entire system shown in Fig. 5.2, noted as the Joint Part of Speech Embedding (JPoSE). The next two sections present experiments using JPoSE on both the fine-grained action retrieval task (using EPIC-Kitchens) and the general video retrieval task (using MSR-VTT). Additionally, the individual PoS-MMENs for the task of Part of Speech Retrieval are tested on EPIC-Kitchens as well as a Part of Speech study is conducted on MSR-VTT to break down the usefulness of different parts of speech.

5.4 Experiments on EPIC-Kitchens

This section contains the experiments using the JPoSE method described in the previous section for the task of fine-grain action retrieval (section 5.4.1) in addition to testing the constituent PoS-MMENs that make up its structure for the tasks of Part of Speech Retrieval (section 5.4.3). A full ablation study of the proposed approach is also conducted (section 5.4.2).

5.4.1 Fine Grained Action Retrieval Results

This section includes the results of JPoSE on the task of fine grained action retrieval on the EPIC-Kitchens dataset. Firstly, details of the methods implementation including feature extraction are given before results on both cross-modal retrieval and within-modal retrieval are presented.

Implementation

Triplet Relevancy Relevancy of triplets for the Part of Speech Multi-Modal Embedding Networks are constructed using the verb and noun classes as discussed in section 5.1. Note that while these classes are generally used for action recognition by way of majority vote, *i.e.* *"put"* is used for actions such as *"place"*, *"put down"* *etc.*, this is only used to determine relevancy. The training and testing of JPoSE is done using the full gamut of open vocabulary verbs and nouns. This means that a video *"place cup"* would be considered relevant to *"put plate"* for the purposes of the verb PoS-MMEN as *"put"* and *"place"* share the same semantic grouping and the textual input representations would differ. To train the Joint Part of Speech Embedding, action level relevancy, as mentioned in section 5.3.3, is defined as both the verb and the noun class having to be identical for the videos/captions to be deemed relevant. Therefore, for each embedding (verb/noun/action), the triplet relevancy sets are distinct.

Video Features Flow and appearance features are extracted from a pre-trained model on Kinetics [60] and fine-tuned on the training set of EPIC-Kitchens[10]. Both networks had been originally trained for the task of action recognition and, in the case of EPIC-Kitchens, used the closed vocabulary clusters. The model was a TSN BNInception model [136] with a separate model trained for appearance and flow features. The features were extracted at the second to last fully connected layer and the resultant appearance and flow features concatenated in order to provide a 2048 dimensional vector which was used as v_i for all experiments.

Text Features Part of Speech tags from the EPIC dataset were determined using the tags discussed in section 5.1. Any other tags that were required were gathered using the large English spaCy parser [3]. Word vectors were extracted using a 100-dimension Word2Vec model trained on the Wikipedia corpus.

Architecture Details Both the verb and noun PoS-MMEN were implemented in the same way. f^k and g^k were coded as a 2 layer perceptron: Two fully connected layers, each with a ReLU layer for non-linearity. The feature inputs and embedded output vectors were L2 normalised. The size of the output space, Ω, for both the verb and noun

[10] The model did not have access to the SEEN or UNSEEN test sets within EPIC.

PoS-MMEN was set to 256, other sizes in the range of 64-1024 were found to have a similar performance. $\hat{f}$ and $\hat{g}$ were implemented as a single shared weights layer and e_v and e_t as the concatenation function — see the ablation study in section 5.4.2 for more information.

Training Details The weighting parameters for the MMEN loss, defined in equation 5.9, were set as follows: $\lambda_{v,t} = \lambda_{t,v} = 1.0$ and $\lambda_{v,v} = \lambda_{t,t} = 0.1$. Similar to [137] the within-modal loss terms, whilst important, were found to be detrimental if given an equal or greater weights than the cross-modal loss terms. This was true for both of the PoS-agnostic losses and the PoS-aware losses. The weights for the final JPoSE loss function, α^{verb} and α^{noun}, were also set to 1 (in equation 5.11). A learning rate of $1e-5$ was used along with an adam optimiser. The action embedding functions $\hat{f}$ and $\hat{g}$ were trained on top of the rest of the network and their weights, $\hat{\theta}$, were initialised via PCA. These hyperparameters were found on initial experimentation with a random 80/20 split of the training set into train/validation sets (with $22,776$ and $5,694$ videos respectively). For all final experiments the full training set was used with no validation set.

Evaluation Metrics All results are reported using the mean average precision metric (mAP) as in the previous chapter. Every video/caption in the test set is treated as a query in turn. For the task of within-modal retrieval the query item (either video or caption depending on if the task is video-to-video or text-to-text) is removed from the test set.

Dataset Both test sets, seen and unseen, of EPIC-Kitchens were used for evaluation. See section 5.1 for further details about the test sets.

Compared Approaches The following approaches are used as comparison to the JPosE learned PoS-MMENs:

- **Random:** Return a random order of items for a given query, included as a lower bound.

- **CCA Baseline:** Canonical Correlation Analysis is used to align both modalities to create an embedding space in which both cross-modal and within-modal retrieval can be performed [40].

- **Features (Word2Vec):** Uses the Word2Vec embedding as the output space for the purposes of ranking items. Only valid for text-to-text retrieval.

- **Features (Video):** Uses the video features as an embedding for the output space for the purposes of ranking items. Only valid for video-to-video retrieval.

- **MMEN(Caption):** A Multi-Modal Embedding Network is trained with the visual features as normal but the textual features are all words from the caption with the word vectors averaged together.

- **MMEN(Verb):** A Multi-Modal Embedding Network is trained with the visual features as normal but the textual features are all the verbs from the caption with the word vectors averaged together.

- **MMEN(Noun):** A Multi-Modal Embedding Network is trained with the visual features as normal but the textual features are all the verbs from the caption with the word vectors averaged together.

- **MMEN(Caption RNN):** A Multi-Modal Embedding Network is trained with g being modelled as an RNN instead of a two layer perceptron. The textual input to the network is the entire caption[11].

- **MMEN([Verb, Noun]):** A Multi-modal Embedding Network is trained with the visual features as normal but the textual features are only the verbs and nouns from the caption. Individually, the verbs and nouns are averaged and the resulting verb/noun representation is concatenated.

Cross-Modal Results

Table 5.2 shows the cross-modal results of JPoSE against the various baselines defined above for both test sets of EPIC-Kitchens. It is noteworthy that, individually, verbs and nouns perform poorly on their own for the task of action retrieval, yet their combination is able to outperform over using the entire caption. This suggests for the task of fine-grained action retrieval these parts of speech are vital to the task, and the inclusion of

[11] NetVLAD [7] was also tested, but was found to have similar results as using an RNN and so was omitted here.

EPIC	SEEN		UNSEEN	
	vt	tv	vt	tv
Random Baseline	0.6	0.6	0.9	0.9
CCA Baseline	20.6	7.3	14.3	3.7
MMEN (Verb)	3.6	4.0	3.9	4.2
MMEN (Noun)	9.9	9.2	7.9	6.1
MMEN (Caption)	14.0	11.2	10.1	7.7
MMEN (Caption RNN)	10.3	13.8	6.3	9.0
MMEN ([Verb, Noun])	18.7	13.6	13.3	9.5
JPoSE(Verb,Noun)	**23.2**	**15.8**	**14.6**	**10.2**

Table 5.2: *Cross-modal action retrieval results on EPIC comparing JPoSE with different MMENs.*

others directly impacts the result.

The addition of the RNN for the text embedding function provides an increase in mAP performance for text-to-video retrieval for both SEEN (+2.6 mAP) and UNSEEN(+1.3 mAP) but sees a larger *decrease* in video-to-text performance. Regardless, the text-to-video retrieval results are still comparable to MMEN([Verb,Noun]).

Whilst disentangling the verbs and nouns from the caption is beneficial over using the full caption, JPoSE, which creates separate embedding spaces for each, outperforms all other baselines for cross-modal retrieval.

Figure 5.3 includes qualitative results of JPoSE against MMEN (caption) and MMEN (caption RNN) [12].

Within-Modal Results

Within-modal results for the task of fine-grained action recognition on EPIC-Kitchens can be seen in Table 5.3. Similarly to cross-modal retrieval, verbs and nouns on their own provide a poor basis for text-to-text retrieval, but perform better than the base video features for the SEEN test set and comparatively for the UNSEEN test set.

[12] A video of the qualitative examples can be seen at: https://www.youtube.com/watch?v=FLS1RQBFow0

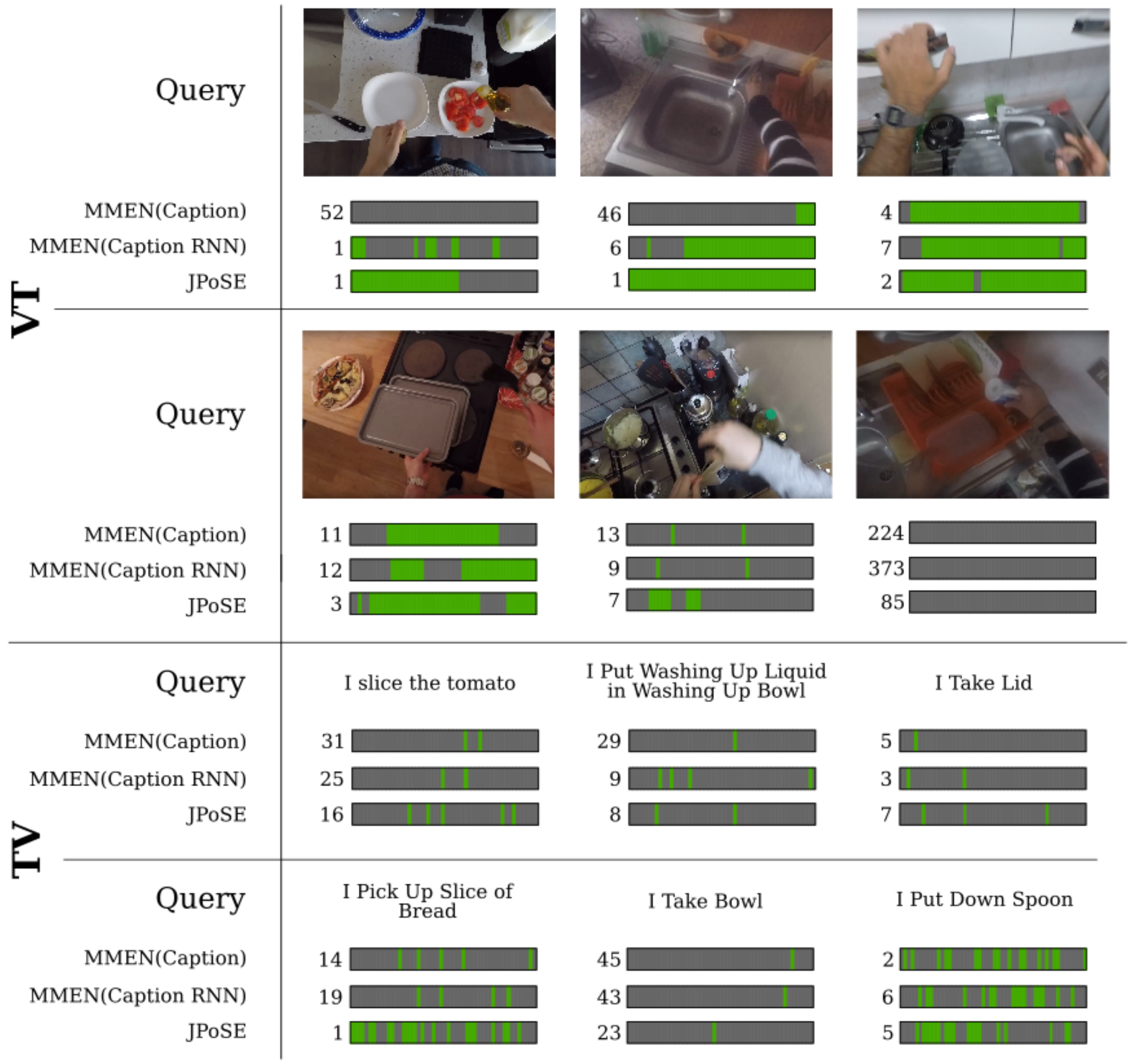


Figure 5.3: *Qualitative cross-modal results for the task of fine-grained action recognition on the EPIC-Kitchens dataset. Both video-to-text (VT, top) and text-to-video (TV, bottom) retrieval results are presented with the query example along with the first 50 retrievals represented by the coloured bar. Green/grey respectively represent a relevant/irrelevant retrieved item. The number in front of each coloured bar shows the rank of the first relevant retrieval (lower rank is better).*

Importantly, the results show how the addition of different modalities is beneficial for within modal retrieval with all methods using captions or the combination of verbs and nouns outperforming the base features on the SEEN dataset. For the within-modal task representing g as an RNN benefits the embedding leading to increases in video-to-video and text-to-text performance over simply using the caption. Regardless, JPoSE outperforms all other methods on both test sets and modality searches.

EPIC	SEEN		UNSEEN	
	vv	tt	vv	tt
Random Baseline	0.6	0.6	0.9	0.9
CCA Baseline	13.8	62.2	18.9	68.5
Features(Word2Vec)	–	62.5	–	71.3
Features(Video)	13.6	–	21.0	–
MMEN (Verb)	15.2	11.7	20.1	15.8
MMEN (Noun)	16.8	30.1	21.2	34.1
MMEN (Caption)	17.2	63.8	20.7	69.6
MMEN (Caption RNN)	17.6	73.5	22.1	76.1
MMEN ([Verb, Noun])	17.6	83.5	22.5	84.7
JPoSE(Verb,Noun)	**18.8**	**87.7**	**23.2**	**87.7**

Table 5.3: *Within-modal action retrieval results on EPIC.*

5.4.2 Ablation Study

In order to understand more about the performance of the JPoSE an ablation study is performed by ablating: the inclusion of the action loss $\hat{L}$ (Eq. 5.11), the encoding functions (e_v, e_t) (Eq. 5.10) and the action embedding functions $\hat{f}, \hat{g}$ (Eq. 5.10). The results can be seen in table 5.4. Of the three encoding functions, concatenate is proven to provide the highest performance over the other two functions whether the action loss is used or not. The action loss provides a marginal benefit, around 1% increase for cross-modal retrieval and only benefiting within-modal retrieval results when the sum and max encoding functions are used. Finally, the addition of the action embedding functions provide a marginal benefit for cross modal retrieval and video-to-video retrieval over the simple combination of the embedding spaces. When comparing the MMEN[Verb, Noun] results from tables 5.2 and 5.3, the largest increase in mAP comes from the two embedding spaces for verbs and nouns over using a single embedding space for actions.

Also shown in the ablation study is the impact of the shared weights between the action embedding functions $\hat{f}$ and $\hat{g}$. On their own, the action embedding functions don't improve the embedding, causing a decrease in mAP over using the concatenated PoS-MMENs. By sharing the weights, JPoSE is able to improve upon using the PoS-aware loss suggesting that the shared weights layer can help align the two modalities in the deeper network.

					EPIC SEEN			
Learn	$\hat{L}$	(e_v,e_t)	$(\hat{f}, \hat{g})$	shared	vv	vt	tv	tt
indep	×	Sum	×	—	17.4	20.7	13.3	86.5
indep	×	Max	×	—	17.5	21.2	13.3	86.5
indep	×	Conc.	×	—	18.3	21.5	14.6	87.1
joint	✓	Sum	(Id, Id)	—	18.1	21.0	14.3	87.3
joint	✓	Max	(Id, Id)	—	18.1	22.4	14.8	87.5
joint	✓	Conc.	(Id, Id)	—	18.3	22.7	15.4	87.6
joint	✓	Conc.	$(\hat{\theta}_{\hat{f}}, \hat{\theta}_{\hat{g}})$	×	17.6	19.3	11.9	85.6
joint	✓	Conc.	$(\hat{\theta}_{\hat{f}}, \hat{\theta}_{\hat{g}})$	✓	**18.8**	**23.2**	**15.8**	**87.7**

Table 5.4: *Ablation study of JPoSE showing the impact of the PoS-aware loss, $\hat{L}$, the encoding functions (e_v, e_t), the action embedding functions $(\hat{f}, \hat{g})$ and, in the presence of the action embedding functions, whether shared weights (shared) were used. Three encoding functions are tested which fuse the outputs of the part of speech embeddings: Sum, Max and Concatenate. The identity matrix (Id) is used to test the absence of the learned functions $\hat{f}$ and $\hat{g}$.*

To further evaluate the differences between the verb and noun embeddings maximum activation examples from the visual functions (f) are shown in Fig. 5.4. For each part of speech, two different neurons are shown and the 9 videos which maximally activate the neurons. Note how similar objects are related in the noun visual embedding function (different items of cutlery and chopping boards) whereas similar actions are grouped together by the verb embedding function (open/close vs. put/take).

5.4.3 Part of Speech Retrieval Results

In this section, the value of using the Joint Part of Speech Embedding for the task of Part of Speech retrieval is evaluated. Specifically, this is using the outputs of the underlying PoS-MMENs (*i.e.* $f^k(v_i)$) instead of the action space ($\hat{f}(\hat{v}_i)$). This tests the notion that by introducing the PoS-agnostic loss, $\hat{L}(\hat{\theta})$, the knowledge of actions can help the performance of the individual part of speech embeddings.

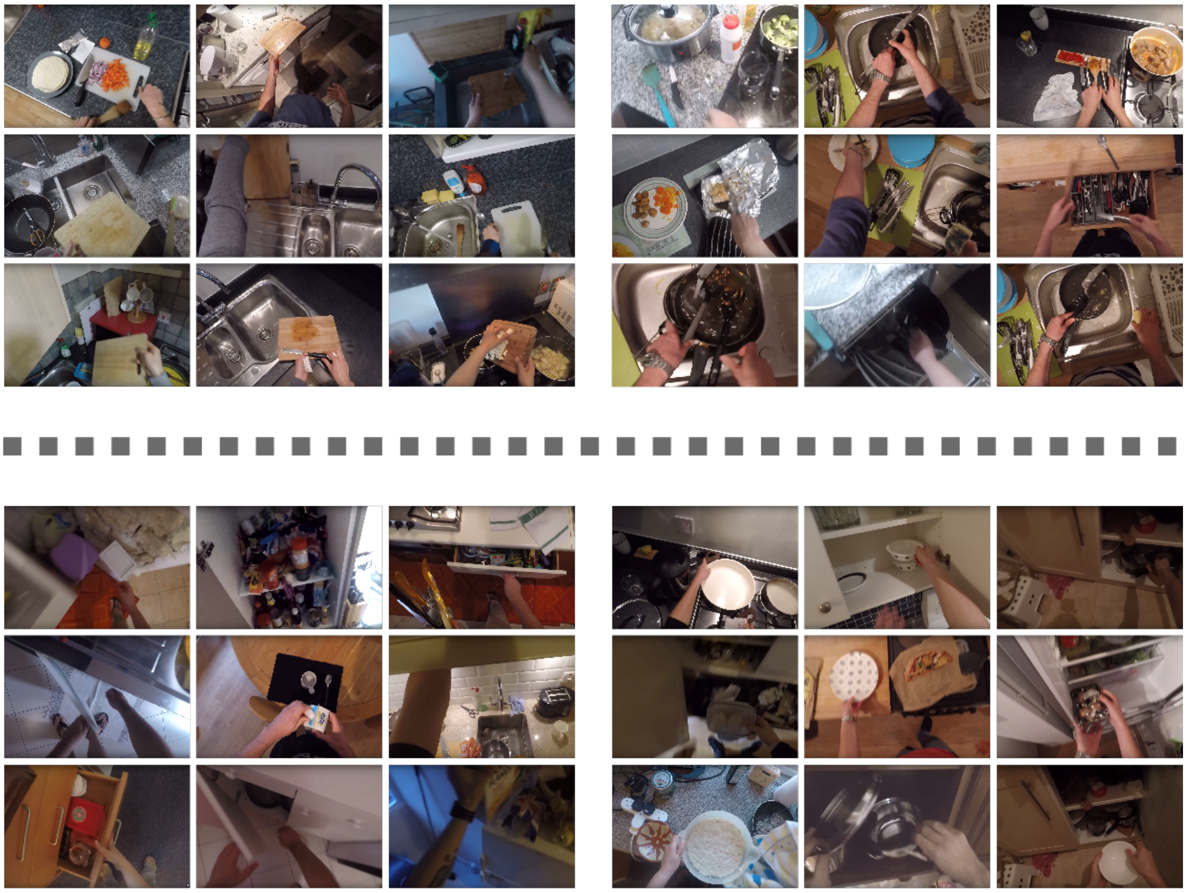


Figure 5.4: *Example maximum activations from two neurons from the noun visual embedding function (top) and the verb visual embedding function (bottom). Examples of similar objects used in different actions can be seen in the noun embedding, chopping boards (left) and cutlery (right). Conversely, in the verb embedding examples of the same action performed on different actions with open/close (left) and put/take (right). A video containing these examples can be seen at* https://youtu.be/FLS1RQBFow0?t=64.

Implementation

The implementation of the JPoSE was exactly the same as in the previous section (section 5.4.1).

Compared Approaches The following approaches are used as comparison to the JPosE learned PoS-MMENs:

- **Random:** Return a random order of items for a given query, included as a lower bound.

- **CCA Baseline:** Canonical Correlation Analysis is used to align both modalities to create an embedding space in which both cross-modal and within-modal retrieval can be performed [40].

- **Features (Word2Vec):** Uses the Word2Vec embedding as the output space for the purposes of ranking items. Only valid for text-to-text retrieval.

- **Features (Video):** Uses the video features as an embedding for the output space for the purposes of ranking items. Only valid for video-to-video retrieval.

- **MMEN(Caption):** A Multi-Modal Embedding Network is trained with the visual features as normal but the textual features are all words from the caption with the word vectors summed together.

- **MMEN([Verb, Noun]):** A Multi-modal Embedding Network is trained with the visual features as normal but the textual features are only the verbs and nouns from the caption. Individually, the verbs and nouns are summed and the resulting verb/noun representation is concatenated.

- **MMEN(Caption RNN):** A Multi-Modal Embedding Network is trained with g being modelled as an RNN instead of a two layer perceptron. The textual input to the network is the entire caption.

- **PoS-MMEN(Verb):** An individual PoS-MMEN is trained with a verb triplet loss (Eq. 5.9 using the verb triplet sets described in section 5.3.2). This is only used for the task of verb retrieval.

- **PoS-MMEN(Noun):** An individual PoS-MMEN is trained with a noun triplet loss (Eq. 5.9 using the noun triplet sets described in section 5.3.2). This is only used for the task of noun retrieval.

Results

Verb Retrieval Results of using JPoSE for verb retrieval are presented in Table 5.5 on the Seen test set of the EPIC-Kitchens dataset (section 5.1). As one would expect, the PoS-MMEN(Verb) performs the best out of the single embedding methods as the task is focused solely on verb retrieval and, as such, any other extraneous information

	EPIC (SEEN)			
	vv	vt	tv	tt
Random Baseline	12.6	12.6	12.6	12.6
Features(Word2Vec)	–	–	–	50.0
Features(Video)	21.0	–	–	–
CCA Baseline	21.3	23.3	25.7	37.7
MMEN(Caption)	32.0	53.1	47.2	90.0
MMEN([Verb,Noun])	33.2	55.7	48.9	96.1
MMEN(Caption RNN)	31.2	33.7	49.2	92.6
PoS-MMEN(Verb)	31.1	56.2	48.5	**97.1**
JPoSE	**33.7**	**57.1**	**49.9**	**97.1**

Table 5.5: *Verb retrieval results on the seen test set of EPIC-Kitchens. Results are given in mean Average Precision (mAP, higher is better).*

from the caption isn't required. Interestingly, it is seen that using the action loss does indeed improve performance by a small amount across all query modalities apart from text-to-text retrieval.

Noun Retrieval Table 5.6 shows the noun retrieval results on the seen test set of EPIC-Kitchens. As with the verb retrieval, knowledge of the actions helps the underlying embedding giving a moderate increase in performance over the base noun PoS-MMEN alone. Additionally, it is clear from comparison that the noun retrieval task is more difficult than the verb retrieval task. The random baseline having a much lower performance for the noun retrieval task highlights the larger number of nouns and noun classes within EPIC-Kitchens compared to verbs and verb classes.

5.4.4 Conclusion of Fine-Grained Action Retrieval Results

For EPIC-Kitchens, in which fine-grained action retrieval was performed, the importance of different parts of speech is paramount. Disentangling the verb and noun from the caption leads to increased performance over using the entire caption. This importance can also be seen in the ablation study that was ran for the proposed method JPoSE. Whilst adding in the action loss and the action embedding functions $\hat{f}$ and $\hat{g}$ give small increases in performance, the largest boost came from simply training the two part of speech embeddings.

	EPIC (SEEN)			
	vv	vt	tv	tt
Random Baseline	2.17	2.17	2.17	2.17
Features(Word2Vec)	–	–	–	30.9
Features(Video)	10.6	–	–	–
CCA Baseline	11.9	16.9	19.2	52.2
MMEN(Caption)	**18.7**	26.2	20.7	70.9
MMEN([verb,Noun])	18.3	29.8	23.8	90.1
MMEN(Caption RNN)	17.9	20.3	22.0	74.0
PoS-MMEN(Noun)	17.8	31.5	23.6	**92.6**
JPoSE	18.6	**32.2**	**25.5**	**92.6**

Table 5.6: *Noun retrieval results on the seen test set of EPIC-Kitchens. Results are given in mean Average Precision (mAP, higher is better).*

Additionally, by learning the underlying PoS-MMENs verb and noun retrieval can be performed allowing for an increased understanding of the embedding space of these different parts of speeches.

5.5 Experiments on MSR-VTT

This section performs experiments of the JPoSE method and part of speech disentanglement for the general video-to-text retrieval task on MSR-VTT. As discussed in section 2.3.2, MSR-VTT is a commonly-used dataset for the task of general video retrieval, representing a good test bed for the generalisability of the notion of Part-of-Speech disentanglement. The adaptions to the main method described in section 5.3.3 are described in section 5.5.1 before results on this task are presented in section 5.5.2. Further results of a part of speech study are presented in section 5.5.3.

5.5.1 JPoSE for General Video Retrieval

For the task of general video retrieval there are two main differences from fine-grained action retrieval as defined in section 5.2: The instance level pairings of cross-modal items and lack of semantic clusters of words are present in the dataset.

For example, in the MSR-VTT dataset [144] for each video there are 20 captions that are considered relevant and for each caption there is a single video that is considered relevant. This leads to cases where *"A cooking tutorial"* and *"A person is cooking"* are considered as (semantically) irrelevant as *"woman is giving a speech on stage"* even though the former two captions share a verb. As discussed in chapter 2, for the general video retrieval task only video-to-text and text-to-video retrieval are performed using the notion of instance-level relevancy, not semantic relevancy, as this is the only defined relevancy within the dataset.

Although text-to-text retrieval could theoretically be performed, as the reasonable assumption that captions that are all relevant to the same video are themselves relevant, it is likely that semantically relevant items could belong to different videos. For video-to-video retrieval on this task, there is no instance level relevancy between videos defined in MSR-VTT apart from the very coarse grained categories which represent very high level information. In this chapter, following from previous works, only results on cross-modal retrieval tasks (*i.e.* video-to-text and text-to-video) are presented.

In order to test the notion of disentangled parts of speech, JPoSE, as defined in section 5.3 is modified as follows:

Firstly, the Mixture-of-Embeddings (MEE) model from miech *et al.* [83] is used as the base for the Multi-Modal Embedding Network (see chapter 2 for further details of their approach). Secondly, the part of speech information can be found using a part of speech tagger from spaCy (similar to EPIC-Kitchens in section 5.1) but there is no higher level semantic information available, *i.e.* there is no way to know which verbs are related to one another. Because of this, the Part of Speech embeddings that make up JPoSE cannot be trained with a PoS-aware loss and so the PoS-agnostic loss is used instead for each of the PoS-MMEN. Note that the input to each PoS-MMEN still only contains the words from its specific part of speech — so a verb-MMEN contains only the verbs from a caption, but will be trained with the PoS-agnostic loss. Finally, as in [83] the bi-directional max-margin ranking loss is used instead of the triplet loss defined in equation 5.9:

$$L(\theta) = \frac{1}{B}\sum_{i}^{B}\sum_{j\neq i} max\big(\gamma + d(f_{v_i}, g_{t_i}) - d(f_{v_i}, g_{t_j}), 0\big) + max\big(\gamma + d(f_{v_i}, g_{t_i}) - d(f_{v_j}, g_{t_i}), 0\big) \tag{5.12}$$

where B is the batch size. This loss performs similarly to the cross-modal triplet loss terms, but, instead of sampling triplets, it compares the relevant item (the corresponding caption in the other modality) with all other irrelevant items in the batch. Thus, it enforces that relevant items are closer together than the average distance between non-relevant items.

5.5.2 General Retrieval Results

Implementation

As mentioned above, the experiments in this section used the code provided by the authors of [83]. The same video, audio and face (where available) features were used to train the embedding. The text features were not used from the authors and were instead generated using the same Word2Vec embedding as used in Section 5.4.1. Part of speech tags were found using SpaCy's large parser model. The original authors use NetVLAD to aggregate the sentences which is tested here against the simple average used in the previous section which was found to outperform it (see the relevant discussion for further information). The encoding functions, (e_v, e_t) were modelled as concatenation.

Compared Approaches

The following approaches are compared to the proposed model:

- **Random:** Return a random order of items for a given query, included as a lower bound.

- **CCA Baseline:** Canonical Correlation Analysis is used to align both modalities to create an embedding space in which both cross-modal and within-modal retrieval can be performed [40].

- **MMEN(Caption):** A Multi-Modal Embedding Network is trained with the visual features as normal but the textual features are all words from the caption with the word vectors summed together.

- **MMEN($< PoS >$):** A Multi-modal Embedding Network is trained with the visual features as normal but the textual features are only the $< PoS >$ from the caption. Multiple Parts of speech tags in square brackets show the results when multiple parts of speech are used as inputs. *i.e.* MMEN([verb, noun]) uses all verbs and nouns from a caption as input the to MMEN.

- **CT-SAN** [147] This method uses a set of tracing LSTMs to discover concept words from video, represented as tracked regions. Semantic attention is applied on top of these representations to create an embedding.

- **JSFusion** [148] Joint pairwise features are calculated between visual features from video frames and word embedding features from captions. This pairwise joint representation is then fed through a convolutional hierarchical decoder to evaluate the similarity between the cross modal items.

- **MEE** [83] This underlying method as is explained in section 5.5.1 with the only difference being the word embeddings. Additionally, the paper only reported text-to-video retrieval results.

Additionally, in the part of speech study, MMEN approaches include NetVLAD to denote when NetVLAD was used to aggregate the sentences and AVG when the average was used.

Evaluation Metrics

The following experiments employ the same evaluation metrics as in the literature. When using the MSR-VTT dataset for the task of video retrieval it is common to use two evaluation metrics, recall@k and median rank.

Recall@k (R@K) shows the recall of first k items returned by a search using the query item. *I.e.*

$$recall@k = \frac{1}{|V|} \sum_{i=1}^{|V|} \frac{top_k(v_i) \cap |T_{i+}|}{|T_{i+}|} \tag{5.13}$$

where $top_k(v_i)$ gives the top k ranked search items for the query v_i. Note that while equation 5.13 is given for video-to-text retrieval this can be modified for text-to-video retrieval by replacing v_i with t_i and T_{i+} with V_{i+}.

However, as it is common to only find a single corresponding caption or video the recall@k for the ith item is simply 1 if the relevant item exists within the top k results or 0 otherwise. The recall@k is then found by averaging over all queries as above.

It is common to set $k = 1, 5, 10$ for the MSR-VTT dataset and larger values of recall@k

	Video-to-text				Text-to-Video			
MSR-VTT Retrieval	R@1	R@5	R@10	MR	R@1	R@5	R@10	MR
CT-SAN [147]	–	–	–	–	4.4	16.6	22.3	35.0
JSFusion [148]	–	–	–	–	10.2	31.2	43.2	13.0
Mixture of Experts [83]*	–	–	–	–	12.9	36.4	51.8	10.0
Random Baseline	0.3	0.7	1.1	502.0	0.3	0.7	1.1	502.0
CCA Baseline	2.8	5.6	8.2	283.0	7.0	14.4	18.7	100.0
MMEN(Verb)	0.7	4.0	8.3	70.0	2.9	7.9	13.9	63.0
MMEN(Caption\Noun)	5.7	18.7	28.2	31.1	5.3	17.0	26.1	33.3
MMEN(Noun)	10.8	31.3	42.7	14.0	10.8	30.7	44.5	13.0
MMEN([Verb,Noun])	15.6	39.4	**55.1**	9.0	13.6	36.8	51.7	10.0
MMEN(Caption)	15.8	40.2	53.6	9.0	13.8	36.7	50.7	10.3
JPoSE(Verb, Noun)	15.5	39.3	53.8	9.0	13.7	37.6	52.2	9.6
JPoSE(Caption\Noun, Noun)	**16.4**	**41.3**	54.4	**8.7**	**14.3**	**38.1**	**53.0**	**9.0**

Table 5.7: *MSR-VTT Video-Caption Retrieval results. For all results apart from Mixture of Experts, the average of 10 runs was taken. *Results are included from [83], only available for Text-to-Video retrieval, but note that MMEN(caption) is identical apart from using a different word2vec embedding.*

are better.

Median Rank (MR) Gives the median rank of the first relevant item returned by a search using a query item. *I.e.*

$$\text{Median Rank} = median(\{rank(v_i), \forall i \in [1, |V|]\}) \tag{5.14}$$

where $rank(v_i)$ returns the rank of the first relevant item to v_i in the search query and $median(\mathcal{X})$ returns the median value of $\mathcal{X}$. Again, Eq. 5.14 is constructed for video-to-text retrieval but by substituting v_i with t_i and V with T it can be used for text-to-video retrieval.

For median rank, lower values of MR are better.

Cross-Modal Results

Table 5.7 shows the general video retrieval results on the MSR-VTT dataset for the task of both video-to-text and text-to-video retrieval. Immediately it is clear that for MSR-VTT verbs are not as important as they are for the task of fine-grained action retrieval, performing much worse for both video-to-text and text-to-video retrieval. Nouns represent a significant source of information within the dataset, providing a large boost in performance in comparison to verbs. This can especially be seen in the results for MMEN(Caption\Noun) which uses the full caption as textual input after removing all the nouns performing much worse than nouns on their own.

In comparison to the fine-grained action retrieval results, using only the verbs and nouns from a caption (MMEN([Verb,Noun])) only gives comparable results to using the full caption. However, this still suggests that for general video retrieval the whole caption isn't necessarily important and that by using only the verbs and nouns similar performance can be achieved. Further discussion about the usefulness of each part of speech can be found in the part of speech study.

The results show two different versions of JPoSE. The first creates separate verb and noun embeddings (denoted by JPoSE(Verb, Noun)) achieving comparable video-to-text retrieval results to MMEN(caption) whilst slightly outperforming it for text-to-video retrieval. This is expected due to the poor performance of MMEN(Verb). From the results above, using verbs by themselves creates a poor embedding space and so it can be expected that, as one of the underlying spaces, the verbs provide little benefit and the performance would suffer as a result. Regardless, this still performs comparably to MMEN(Caption) for video-to-text retrieval and slightly better for text-to-video retrieval.

The final version of JPoSE, denoted by JPoSE(Caption\Noun, Noun), creates an embedding space with all parts of speeches without nouns along with the noun embedding space. As from the MMEN results Caption\Noun proved to be much more informative than verbs, and as such creates a better space for general video retrieval than JPoSE(Verb,Noun).

Figure 5.5 shows some qualitative results of the proposed method JPoSE(Caption\Noun, Noun) against MMEN(Caption). JPoSE is able to frequently rank videos for a caption higher than the baseline approach.

5.5.3 Part of Speech Study

Given the results in Table 5.7 it was deemed important to evaluate the importance of the different parts of speech within the MSR-VTT dataset. The top 5 parts of speech

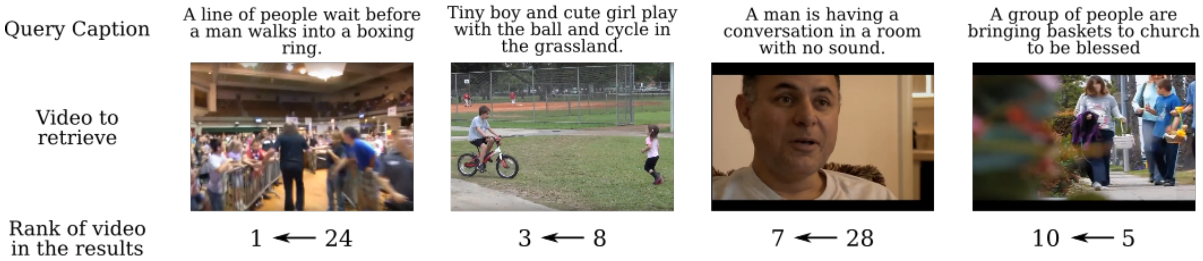

Query Caption
A line of people wait before a man walks into a boxing ring.
Tiny boy and cute girl play with the ball and cycle in the grassland.
A man is having a conversation in a room with no sound.
A group of people are bringing baskets to church to be blessed
Video to retrieve
Rank of video in the results
1 ⟵ 24
3 ⟵ 8
7 ⟵ 28
10 ⟵ 5

Figure 5.5: *Qualitative text-to-video results of action retrieval on MSR-VTT. A ← B shows the rank B of the retrieved video from using the full caption MMEN(caption), equivalent to MEE [83], and the A the rank of the same video when disentangling the caption using the proposed method JPoSE(Caption\Noun, Noun).*

for MSR-VTT were tested in different combinations. The results can be seen in table 5.8.

Average vs. NetVLAD For MSR-VTT using NetVLAD provides a considerable benefit over simply averaging the word representations for all experiments but MMEN(Verb). This is in stark comparison to the EPIC-Kitchens results which can be explained due to the difference in length of captions. Due to the much longer captions in MSR-VTT the NetVLAD aggregation is much more important than simple averaging, outperforming it greatly[13].

Individual Part of Speech Results The results show that, on their own, parts of speech other than verbs and nouns provide very little discriminative information for the embedding. Verbs and nouns perform much better relatively, but are still beaten by using the whole caption.

Combining Different Parts of Speech Combining the verbs and nouns together leads to a boost in performance compared to using the individual parts of speech, especially when using NetVLAD for aggregation. Determiners add little to the embedding, providing a drop in accuracy. Considering the determiners present in MSR-VTT this can be expected: *"the"* and *"a"* add little to understanding and discriminating between captions.

[13] For EPIC-Kitchens, the opposite was found, likely due to the much shorter captions.

However, by adding in the adverbs and adjectives to the embedding, small increases in performance can be seen which becomes comparable to using the entire caption for both video-to-text and text-to-video retrieval.

Caption Disentangling with JPoSE Table 5.8 also includes the results using JPoSE(Caption\Verb, Verb). It could be theorised that due to the much lower performance of verbs compared to nouns for an MMENthat this verb disentanglement would lead to much worse results than JPOSE(Caption\Nouns, Noun) but it actually achieves higher recall@10 and a lower median rank for video-to-text retrieval. Otherwise, it seems to perform comparably to JPoSE(Verb, Noun).

5.5.4 Conclusion of General Video Retrieval Results

Whilst JPoSE doesn't provide the huge gain in retrieval performance for the task of General Video Retrieval that it did for Fine-Grained Action Retrieval by disentangling the caption using knowledge of the different parts of speech can lead to an increase in the final results. It can also be theorised that with the semantic knowledge of how the verbs and nouns in MSR-VTT are related (or indeed other parts of speech) JPoSE could see a larger increase in performance.

The importance of disentangling the caption as well as different parts of speech is clear from the results. As with EPIC-Kitchens, using the whole caption isn't necessarily important — determiners particularly cause a drop in performance for retrieval.

5.6 Conclusion

This chapter has presented work on understanding different parts of speeches, indeed expanding the scope of previous chapters which focus solely on verbs, for a large-scale dataset EPIC-Kitchens. It has shown that disentangling the caption into its constituent parts of speech can provide a gain in performance for both the task of fine-grained action retrieval as well as the general video retrieval task.

Whilst multi-verb annotations could not be collected for EPIC-Kitchens due to its size, the clustering of verbs and nouns allows for the method to intrinsically learn which words are semantically similar and construct a relevant embedding space. Accordingly, an approach that factored in this knowledge was created by learning separate part of speech embeddings, themselves useful for tasks where only the act or object are required, with a final action embedding learned on top.

This approach was tested for two different retrieval tasks, the fine-grained action retrieval task on EPIC-Kitchens which includes shorter captions/videos and the general video-retrieval task on MSR-VTT with comparatively longer videos and captions.

The importance of disentangling the caption was clear to see from the results whereby only using certain parts of speech using a standard cross-modal embedding technique gave comparable or higher performance than using the entire caption. The proposed method increased results even further, not only for action retrieval but also showing that by including knowledge of the entire action then the underlying part of speech embeddings can be improved.

The proposed approach does simplify the intra part-of-speech relationships however, with the word vectors being combined via averaging. For example, *"put bowl in pan"* would have the same representation as *"put pan in bowl"*. For EPIC-Kitchens, the captions are short and generally simple enough that this doesn't prove much of a problem. However, modelling the relationships within each part of speech (and indeed explicitly doing so between different parts of speech) is an obvious extension of the work.

Finally, while the method can be used to predict multiple verbs for an action this is somewhat limiting compared to the multi-verb representations explored in chapter 4. Only main verbs can be predicted due to the absence of contextual clues for supplementary verbs. Another extension of the method would be to try to capture this information using a similar clustering method on EPIC but for supplementary verbs.

	Video-to-text				Text-to-Video			
MSR-VTT Retrieval	R@1	R@5	R@10	MR	R@1	R@5	R@10	MR
CT-SAN [147]*	–	–	–	–	4.4	16.6	22.3	35.0
JSFusion [148]*	–	–	–	–	10.2	31.2	43.2	13.0
Mixture of Experts [83]*	–	–	–	–	12.9	36.4	51.8	10.0
Random Baseline	0.3	0.7	1.1	502.0	0.3	0.7	1.1	502.0
CCA Baseline	2.8	5.6	8.2	283.0	7.0	14.4	18.7	100.0
MMEN(DET AVG)	0.0	0.2	0.5	214.0	0.3	1.0	2.2	264.0
MMEN(ADJ AVG)	0.0	0.3	0.7	216.0	0.1	1.1	2.6	260.0
MMEN(ADP AVG)	0.1	0.6	1.5	172.0	0.7	2.8	5.0	185.0
MMEN(Verb AVG)	1.1	5.4	11.1	57.0	3.2	10.9	17.4	57.0
MMEN(Noun AVG)	10.0	28.0	40.0	16.0	10.7	29.7	43.5	15.0
MMEN(DET NetVLAD)	0.0	0.1	0.3	241.0	0.1	1.1	2.4	255.0
MMEN(ADJ NetVLAD)	0.0	0.0	0.1	232.0	0.2	1.2	2.0	262.0
MMEN(ADP NetVLAD)	0.1	0.7	1.5	174.0	0.6	2.9	4.9	190.0
MMEN(Verb NetVLAD)	0.7	4.0	8.3	70.0	2.9	7.9	13.9	63.0
MMEN(Noun NetVLAD)	10.8	31.3	42.7	14.0	10.8	30.7	44.5	13.0
MMEN([V, N, DET] AVG)	9.0	28.4	41.0	15.0	7.7	24.2	36.0	20.0
MMEN([V, N] AVG)	12.9	34.0	46.7	12.0	12.6	32.6	46.3	12.0
MMEN([V, N, ADP] AVG)	13.0	33.0	46.0	13.0	12.2	33.0	46.0	13.0
MMEN([V, N, ADJ] AVG)	12.4	32.9	45.3	13.0	11.0	31.2	44.3	13.0
MMEN([V, N, ADJ, ADP] AVG)	13.0	32.3	45.9	12.0	11.1	31.5	44.3	13.0
MMEN([V, N, DET] NetVLAD)	14.8	38.3	52.5	9.1	12.4	33.6	46.3	13.0
MMEN([V, N] NetVLAD)	15.6	39.4	55.1	9.0	13.6	36.8	51.7	10.0
MMEN([V, N, ADP] NetVLAD)	15.8	40.3	55.1	8.5	13.8	36.7	51.0	10.0
MMEN([V, N, ADJ] NetVLAD)	16.3	40.1	54.1	8.9	14.0	36.2	50.9	10.0
MMEN([V, N, ADJ, ADP] NetVLAD)	16.1	39.7	53.8	8.9	13.4	36.2	51.3	10.0
MMEN(Caption AVG)	12.4	32.8	45.6	12.0	11.4	31.2	43.8	14.0
MMEN(Caption NetVLAD)	15.8	40.2	53.6	9.0	13.8	36.7	50.7	10.3
JPoSE(Verb, Noun)	15.5	39.3	53.8	9.0	13.7	37.6	52.2	9.6
JPoSE(Caption\Verb,Verb)	15.9	39.2	**55.5**	**8.0**	13.4	36.8	52.0	10.0
JPoSE(Caption\Noun,Noun)	**16.4**	**41.3**	54.4	8.7	**14.3**	**38.1**	**53.0**	**9.0**

Table 5.8: *Part of Speech study on the MSR-VTT dataset using verbs (V) nouns (N) adjectives (ADJ) adpositions (ADP) and determiners (DET). A comparison between using NetVLAD and simply averaging the word representations was evaluated. Results calculated using recall@k (R@k, higher is better) and median rank (MR, lower is better). For each row, an average of 10 runs is reported. *Results are included from [83, 147, 148], only available for Text-to-Video retrieval.*

Chapter 6

Zero Shot Recognition and Retrieval Using an Open Vocabulary

A zero-shot task is one in which classes are present in the testing set that aren't in the training set. Naturally, this makes for a very challenging problem as it can be difficult to train models to successfully predict and recognise unseen classes. This chapter will present two tasks focused around the notion of performing zero-shot recognition or retrieval.

Firstly, the contextual annotations from chapter 4 will be used to perform zero-shot recognition of verbs. Due to the multi-label nature of the collected annotations classes which haven't been used in training can be expressed using verbs which have been seen. This emulates how a human would describe an unseen task. For example, given an action that someone has never seen before, such as *"poaching"*, the words *"cook"*, *"boil"* or *"simmer"* might be used to describe this cooking technique.

Secondly, the disentangling of various Parts of Speech in the method from chapter 5 can be used to perform zero-shot retrieval for unseen (combinations of) classes. *I.e.* If the verb *"put"* has been seen but the noun *"saucer"* hasn't, the task is to evaluate whether the caption *"put saucer"* be correctly retrieved?

The makeup of this chapter will be as follows: The definition of a zero-shot task will be given in section 6.1 where different forms will be introduced. Next, section 6.2 will present results for Zero-Shot Action Recognition wherein no visual or textual knowledge is used during training. Finally, in section 6.3, zero-shot experiments for fine-grained action retrieval on EPIC-Kitchens will be presented.

6.1 Zero-Shot Tasks

This section will introduce the concept of a zero shot task, how it differs from the standard many-shot approach and the different ways that zero-shot tasks can be defined. Then, it will describe the two tasks which will be used in this chapter. Specifically, the definition of zero-shot tasks, in comparison to many-shot and few-shot, will be presented in section 6.1.1. Using verbs as attributes will be presented in section 6.1.2 and zero-shot using semantic information will be introduced in section 6.1.3.

6.1.1 Many vs. Few vs. Zero Shot

Classically, machine learning tasks are constructed using two distinct sets of data which form a training set and a testing set. The training set consists of input data used to train a model which, via being discriminative or generative, learns aspects of the underlying data. The testing set is utilised solely with the aim of evaluating how well the model was able to learn and generalise on new instances of data.

Whilst the training and test sets are distinct in terms of instances, the classes present are traditionally kept the same, *i.e.* There will be a *"put pan"* class in both the train and test sets but different videos with the class label will be present in each set. The difference between many, few and zero-shot can then be explained by the number of instances a class has in the training set compared to the test set.

Many Shot, which is the standard set-up, implies that there are 'many' training examples available for each class. This can be in the hundreds, thousands or much more. Comparatively, in a *Few Shot* learning scenario, one or more classes in the training set only contain a 'few' examples[1]. Many datasets which use open vocabulary annotations have a long-tailed distribution. Because of this class imbalance, some classes will be few-shot classes.

[1] Of course, these definitions aren't strict and, in most cases, are defined in relative terms. Regardless, few-shot learning as a term can generally be used to describe training when classes contain only a handful of examples, *e.g.* < 10 examples.

Verb	# in Train	# in Test	Type
Put	2,930	1,380	Many Shot
Open	2,785	949	Many Shot
Pick-Up	2,577	943	Many Shot
Take	2,258	917	Many Shot
Fill	90	29	Few Shot
Check	89	31	Few Shot
Put-On	86	37	Few Shot
Spoon	73	29	Few Shot
Shake-Off	0	25	Zero Shot
Rinse-Off	0	17	Zero Shot
Clean-Off	0	6	Zero Shot
Push-Down	0	5	Zero Shot

Table 6.1: *Examples of Many Shot, Few Shot and Zero Shot verbs in EPIC-Kitchens.*

Zero-shot learning is the complete absence of any examples in the training set: the class only appears in testing for evaluation. This represents a very challenging problem, but one that can be common in practice. With CNNs becoming core techniques for computer vision, the amount of data required during training is leading to large datasets. However, not every class can be represented or collected during training. This can be especially true when using an open vocabulary to describe or define classes. Given the long-tailed nature of collected annotations it becomes increasingly more challenging to collect all possible examples and train a model which is able to learn the examples effectively (*i.e.* not overfit on the more numerous classes).

Table 6.1 includes examples of many-shot, few-shot and zero-shot classes from the EPIC-Kitchens dataset. For EPIC-Kitchens, a verb or noun class was considered a many shot class if there existed 100 or more different instances in the training set[2]. Due to the open vocabulary the verbs and nouns within EPIC-Kitchens exhibit a long-tailed distribution. This causes examples of all three types of classes (many-shot, few-shot and zero-shot) as described above.

[2] Action classes, which are created as the cross product between verbs and nouns, can therefore have less than 100 instances.

Generalised Zero-Shot

Normally, zero-shot tasks use a test set created solely of zero-shot classes. However, Chao *et. al* [16] introduce the idea of generalised zero-shot in which the test set includes both many or few shot classes (*i.e.* classes in which there are examples in the training set) and zero-shot classes (*i.e.* classes in which there are no examples in the training set). This construction for the test set can be argued to be much more realistic for computer vision applications, but, importantly forces a method to not only predict between novel classes; it also needs to evaluate whether the instance belongs to a class that has been seen before during training or not. The experiments in the previous chapter (chapter 5) are all examples of generalised zero-shot experiments. As the many-shot examples were used for training there existed some examples in both the seen and unseen test sets which were present in the training set and some which weren't. This will be further explored in section 6.3.

6.1.2 Describing Unseen Actions with Multiple Verbs

The multi-verb, verb-only representations from chapter 4 show a clear benefit for zero-shot recognition. Firstly, by using only verbs, the labelling representation is object agnostic, meaning that if a video contains an object not seen in training then only the action itself (manner/result) needs to be recognised. This is highly beneficial in cases where the actions between objects are very similar. For example, if *"bowl"* wasn't seen in training but *"plate"* had been the motions of *"put[ting]"*, *"clean[ing]"* or *"pick[ing] up"* would be similar regardless of the shape of the object. Of course, this isn't always the case, but as long as similar objects have been seen in training, it is possible for the action to still be found.

Secondly, verbs are being learned, not classes. In this way, verbs can be thought of as the attributes that describe and differentiate between the different actions. So, by introducing a new class the same verbs that have been learned can be used to describe the unseen class. For example, even though the class *"turn-off burner"* might not have been seen during training, the model can predict verbs to try and describe the unseen class. Verbs such as *"rotate"* or *"hold"* represent correct predictions for this class for similar actions that have been seen during training.

6.1.3 Zero-Shot Using Semantic Information

Most methods which attempt zero-shot use semantic information as additional input during training. For example, for the caption *"put pan on the hob"* which, along with a corresponding video, is not present in the training set, then knowledge of the words *"put"*, *"pan"* or *"hob"* can be used to work out what the caption could mean even though the example has not been seen. Generally, this has been achieved through the use of unsupervised word embeddings [85] (such as in [47]), which allow for unseen classes to be relatable to seen classes through the use of the learned similarity scores. For example, the noun *"saucepan"* might not have been seen during training, but *"pot"* has been seen, then, as the similarity score between the two words is high (0.71), it can be expected that in a learned embedding space *"pot"* would be embedded near *"saucepan"*. Note that here both words exist in the corpus that was used to originally train the unsupervised word embedding but no visual examples for *"pot"* would have been seen during training[3].

This task is utilised in section 6.3 for the EPIC-Kitchens dataset using the method JPoSE from chapter 5. As the input to the text modality is represented using word vectors from Word2Vec, then the information from the unsupervised training, along with the part of speech disentanglement, can be used to help retrieve zero-shot instances.

6.2 Zero-Shot Action Recognition Using Multiple Verbs

In this section, zero-shot results will be presented on GTEA Gaze+ using the contextual annotations collected in chapter 4. Verbs will be treated as attributes that describe the action taking place, allowing for unseen *action classes* (which can be thought of as unseen combinations of verbs) to be predicted. Experiments will be performed on three different train/test splits of GTEA Gaze+ which show the effect of a decreasing number of training examples and how the different verb-only labelling representations are affected.

[3] This task can be likened to how humans can read about an exotic animal they have never seen before and then visually recognise a picture of it from description alone.

% split	# Action Classes	# Videos	# SV Verbs	# MV Verbs	# SAMV Verbs
90/10	31/3	886/115	14/3	32/12	90/66
80/20	27/7	780/221	13/5	19/19	90/77
50/50	17/17	506/495	11/11	28/25	83/88

Table 6.2: *Details about the three different train/test splits of GTEA Gaze+ used for the zero-shot experiments. #/# represents number in the train and test splits respectively.*

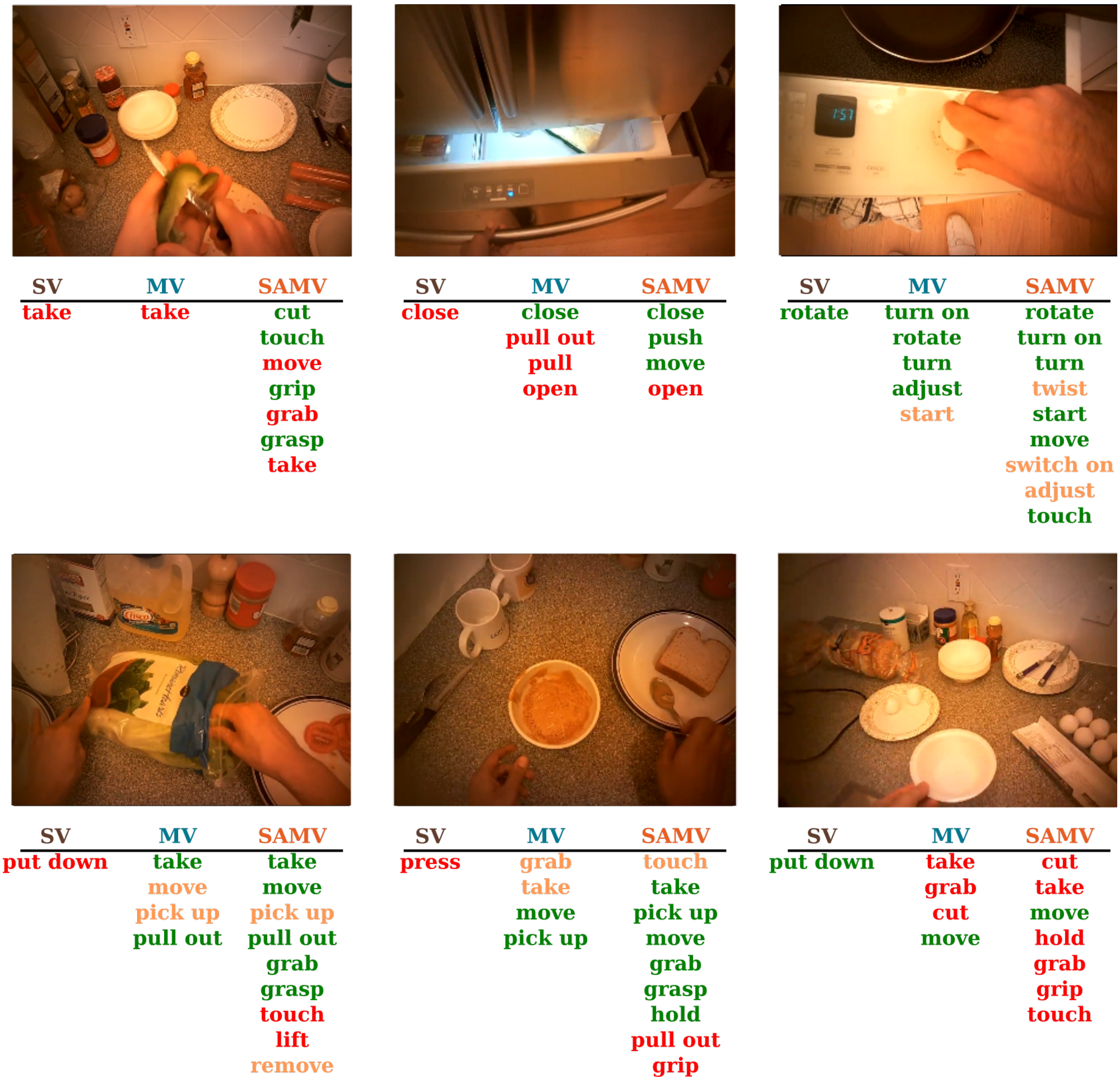
6.2.1 Zero-Shot Test Sets of GTEA Gaze+

GTEA Gaze+ was chosen for the zero-shot experiments as it was the largest of three datasets tested in chapter 4. As no zero-shot test sets for GTEA Gaze+ were available, three were created with differing percentages of unseen action classes (similar to [47]). This allowed for the evaluation of the different models on their generalisability when the number of zero-shot action classes increases and the number of training examples decreases.

Three different train/test splits were formulated based on the percentage of unseen action classes in each test set. To test the impact of a decreasing number of training examples, the percentage of unseen action classes in the test sets were made with {10%, 20%, 50%} of action classes in the test split. In this way, the test sets were created by randomly sampling n% action classes from the overall datasets, the other action classes were then used as the training set. Because of the action class imbalance within GTEA Gaze+, the test sets were sampled such that the number of videos in the test set is similar to the percentage of action classes in the test set (this was to prevent cases where 20% of videos could be sampled from 10% of the action classes *etc.*).

Table 6.2 shows the sizes of the three train/test splits for GTEA Gaze+ in terms of number of action classes within the zero-shot testing sets as well as the number of videos. Included below is the list of zero-shot action classes for each train/test split.

- **90/10** *"close freezer", "turn on burner", "put knife".*
- **80/20** *"put plate", "take knife", "turn-off burner", "put cup", "open microwave", "put spoon [on] plate", "turn-off tap".*
- **50/50** *"take lettuce", "cut pepper [using] knife", "close fridge", "put plate", "crack egg [into] bowl", "put tomato", "take spoon [from] plate", "open freezer", "put cup",*



SV	MV	SAMV
take	take	cut
		touch
		move
		grip
		grab
		grasp
		take

SV	MV	SAMV
close	close	close
	pull out	push
	pull	move
	open	open

SV	MV	SAMV
rotate	turn on	rotate
	rotate	turn on
	turn	turn
	adjust	twist
	start	start
		move
		switch on
		adjust
		touch

SV	MV	SAMV
put down	take	take
	move	move
	pick up	pick up
	pull out	pull out
		grab
		grasp
		touch
		lift
		remove

SV	MV	SAMV
press	grab	touch
	take	take
	move	pick up
	pick up	move
		grab
		grasp
		hold
		pull out
		grip

SV	MV	SAMV
put down	take	cut
	grab	take
	cut	move
	move	hold
		grab
		grip
		touch

Figure 6.1: *Qualitative zero-shot results on GTEA Gaze+ using verbs as attributes for $\phi_{\{SV,MV,SAMV\}}$. Green verbs represent correct predictions, red verbs show incorrect predictions and verbs in orange denote that the verb was predicted at a much higher/lower rank in comparison to the ground truth.*

"put bowl", "turn-on burner", "take cup", "put spoon [on] plate", "take bowl", "take pepper [from] bowl", "put bread", "turn-on tap", "turn-off tap"

6.2.2 Results of Zero-Shot Action Recognition Using Context

The results of using $\phi_{\{SV,MV,SAMV\}}$ on the three different zero-shot train/test splits can be seen in table 6.3 which includes both accuracy (calculated using eq. 4.9) as well as

% split	90/10	80/20	50/50
SV	32.2	14.9	13.1
MV	33.2	42.9	25.6
SAMV	**54.8**	**44.4**	**30.6**

% split	90/10	80/20	50/50
SV	39.0	13.8	19.3
MV	39.6	**41.3**	36.8
SAMV	**61.4**	40.1	**41.7**

Table 6.3: *Table of zero-shot results using the three verb-only labelling methods from chapter 4 on the constructed zero-shot datasets. Left: Results are shown using accuracy with* $\alpha = 0.3$ *(eq. 4.9). Right: Results are shows using per-class accuracy with* $\alpha = 0.3$*. (*α *is the threshold for which* V_i^{SAMV} *is created — see equation 4.9 for more details.).*

per class accuracy (calculated using the same equation per action class and averaged over all action classes). The implementation of ϕ was the same as in section 4.8 (*i.e.* a sigmoid cross entropy loss, eq 4.2, was used for $\phi_{\{MV,SAMV\}}$).

As expected, the zero-shot models achieve lower accuracies than the results on the normal dataset which decrease as the number of training examples decreases. Due to the expanded vocabulary of Soft-Assigned Multi-Verb (SAMV), it is able to consistently outperform the Multi-Verb (MV) representation across all train/test splits of GTEA Gaze+. This is true even when the number of verbs in the test set is larger than those present in the training set (in the case of the 50/50 set for SAMV, see table 6.2).

The Single-Verb representation has the lowest accuracy across all train/test splits seeing a large drop in performance between using 90% and 80% of the dataset for training. This can be explained by the limited vocabulary that the Single Verb (SV) representation learns in comparison to the other labelling methods. In addition, under SV, each video is only assigned a single verb thus each video only has a single attribute making it extremely difficult for a model to perform zero-shot recognition.

Figure 6.2 shows the results of ϕ_{SAMV} for different values of α (see section 4.8.1). Whilst the difference between the 90/10 split and the 80/20 split is very slight, there is a large drop in accuracy across all values of α for the 50/50 split showing its difficulty. This can also be explained by looking at the distributions of verbs in the training and test splits. Within both the 90/10 and the 80/20 split, all 90 verbs are seen in the training set, meaning no new attributes are present in the test set. Comparatively, the 50/50 split contains only 83 verbs in its training split (table 6.2) and includes 7 unseen verbs in the testing split.

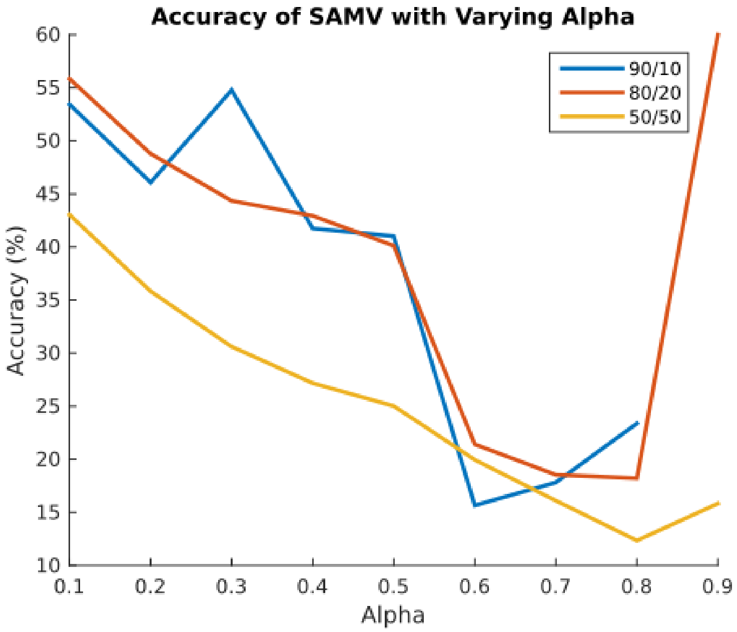


Figure 6.2: *Accuracy of ϕ_{SAMV} on the three zero-shot train/test splits of GTEA Gaze+. Note that the 90/10 test split has 0 verbs with a value of* 0.9 *or higher and accordingly the accuracy at* $\alpha = 0.9$ *is undefined. A large peak for the 90/10 split at* $\alpha = 0.3$ *can be seen. This can be explained by the large reduction of verbs within the smallest of the three test splits. This also explains the large peak in the 80/20 split at* $\alpha = 0.9$.

From looking at high values of α[4] (comparable to the SV labelling representation) and $\alpha = 0.5$ (similar to MV) the efficacy of ϕ_{SAMV} can be compared to ϕ_{SV} and ϕ_{MV}. For the 90/10 and 50/50 splits ϕ_{SAMV} outperforms ϕ_{SV} by a moderate amount. For the 80/20 split there is a sharp peak at $\alpha = 0.9$, though there are only 10 videos which contained a verb with a value of 0.9 or higher. Even when comparing $\alpha = 0.8$ ϕ_{SAMV} beats ϕ_{SV}.

When looking at the comparison between $\alpha = 0.5$ for ϕ_{SAMV} to ϕ_{MV} a different story emerges: a gain of +8% is seen for the 90/10 split, a drop of 2.8% for the 80/20 split as well as a moderate drop in accuracy of -0.6% for the 50/50 split. This suggests that the addition of the supplementary verbs (supplementary verbs are those which have a low assignment score, see figure 4.6 for more details) isn't as important as the inclusion of more main verbs for zero-shot tasks and, in some cases, can be a mild hindrance when using the verbs to describe the novel action classes.

[4] α is the threshold for which V_i^{SAMV} is created — see equation 4.9 for more details.

The Multi-Verb representation achieves similar performance to using a single verb for the 90/10 split of unseen action classes and, indeed, shows an increase in accuracy when fewer training action classes are used (for the 80/20 split). In addition, the MV representation achieves consistent per-class accuracy across all splits showing good generalisability across differing amounts of training classes.

Figure 6.1 shows qualitative results of $\phi_{\{SV,MV,SAMV\}}$ on the zero-shot 50/50 train/test splits of GTEA Gaze+. Using single verb labels often leads to incorrect and non-sensical verbs being predicted whereas using a Multi-Verb or Soft-Assigned Multi-Verb representation gives a higher number of correct verbs being predicted. However, this still represents a challenging problem for the unseen action classes in that antonyms can be seen together on a few of the examples such as {*"Open"*, *"Close"*} and {*"take"*, *"put"*} being confused.

Whilst some of the incorrect verbs that $\phi_{\{MV,SAMV\}}$ predict represent logical descriptions (for example *"touch"*, *"grip"* or *"grab"*) there is also a large amount of confusion between common actions which occur in similar locations using similar objects, *e.g.* *"cut"* is predicted by both $\phi_{\{MV,SAMV\}}$ for *"put bowl"* (lower right in figure).

6.2.3 Conclusion Zero-Shot Action Recognition Using Context

The multi-verb representations from chapter 4 can be thought of as attributes for the different action classes present in action recognition. Because of this, combinations of seen verbs can be used to describe unseen actions, allowing for a relatively small vocabulary of 90 verbs for training.

This notion was tested and evaluated on the GTEA Gaze+ dataset, representing the largest dataset in which the multi-verb annotations were collected. Additionally, the reliance on training examples required to successfully perform zero-shot recognition was assessed with varying levels of training data being provided to the models.

Overall, the usage of Soft-Assigned Multi-Verb labels gave the best results at predicting correct verbs used to describe the novel actions. Of particular note, is the performance of the model trained with the hard-assigned Multi-Verb labels, which lead to consistent per-class accuracy results even when the ratio of training examples to testing examples was decreased.

6.3 Zero-Shot for Fine Grained Action Retrieval

In this section Zero-Shot results will be presented for the EPIC-Kitchens dataset using the JPoSE method originally introduced in chapter 5. This will be using the combination of disentangling the caption into its constituent parts of speech in addition to the prior unsupervised knowledge from textual corpora (see section 6.1.3).

6.3.1 Zero Shot Classes in EPIC-Kitchens

The open vocabulary captions of EPIC-Kitchens naturally lead to zero-shot classes in both the seen and unseen test sets (see sections 5.1 and 6.1.1 for more details). Because of this, the results presented in the previous chapter were in fact set up as a Generalised Zero-Shot scenario.

Due to EPIC-Kitchens being constructed as the cross product of individual verb and noun classes, there are three different types of captions which represent zero-shot captions. Firstly, the noun within the caption is present within the training set but the verb has not is given the name *Zero-Shot Verb*. Secondly, the verb is present during training but the noun is not, defined as *Zero-Shot Noun* (or ZSN). Finally, if both the verb and the noun are in the training set then the entire action is novel, representing a very challenging, but sparse task with very few examples (see table 6.4). The former two test sets will be evaluated later in this section.

Table 6.4 shows the counts of the number of instances for each of the three settings introduced in the previous paragraph for EPIC-Kitchens. Noteworthy, is that a total of 12% of the total videos over both original tests sets (Seen and Unseen) represent zero-shot instances. Additionally, there are a similar number of of ZSN instances to the number of ZSV instances (709 *vs.* 642). Comparatively, the number of zero-shot actions is much lower with only 126 different videos comprising of 62 unique actions across both test sets[5].

[5] Indeed, the zero-shot actions are completely disparate between the seen and unseen test sets.

EPIC	All				ZSV		ZSN		ZSA	
	Videos	Verbs	Nouns	Actions	Videos	Verbs	Videos	Nouns	Videos	Actions
Train	26,710	192	1005	2,513	–	–	–	–	–	–
Seen	8,047	232	530	1,241	452	119	367	80	64	29
Unseen	2,929	136	243	634	257	63	275	127	62	33

Table 6.4: *Number of Videos and verbs/nouns in the three splits of EPIC-Kitchens. Additionally, number of videos and zero-shot classes in Zero-Shot Verbs (ZSV), Zero-Shot Nouns (ZSN) and Zero-Shot Actions (ZSA) are also shown.*

6.3.2 Zero-Shot Results on EPIC-Kitchens

Table 6.5 includes the results of JPoSE from chapter 5 along with the following baselines from the same chapter:

- **Random** Included as a lower bound, the rankings of retrieved items for each query are randomised.

- **CCA** Canonical Correlation Analysis (or CCA) involves learning a weights matrix which matches the correlation between two sets of items.

- **MMEN(Caption)** This uses a Multi-Modal Embedding Network (MMEN, see section 5.3.1 for more details) trained using the entire caption aggregated using simple averaging.

- **MMEN(Caption RNN)** In this case the Multi-Modal Embedding Network is trained using the entire caption, but the word vectors are aggregated via the use of an RNN.

From the results, the importance of disentangling the caption can be seen, especially for the cases where the captions represent the zero-shot combinations (*i.e.* ZSV and ZSN), with JPoSE outperforming all other baselines. Interestingly, MMEN(Caption RNN) generally shows incremental improvements over MMEN(Caption) suggesting that the addition of the GRU can be more beneficial for zero-shot tasks.

Figure 6.3 shows qualitative zero-shot results on EPIC-Kitchens comparing the two baselines to JPoSE. Using JPoSE leads to not only lower ranks of the first relevant retrieval, but also other relevant retrievals are consistently ranked higher than the baseline methods.

EPIC	ZSV		ZSN	
	vt	tv	vt	tv
Random Baseline	1.57	1.57	1.64	1.64
CCA Baseline	2.92	2.96	4.36	3.25
MMEN (Caption)	5.77	5.51	4.17	3.32
MMEN (Caption RNN)	4.83	6.01	4.43	4.28
JPoSE	**7.50**	**6.47**	**7.68**	**6.17**

Table 6.5: *Results of using JPoSE for Zero-Shot Verbs (ZSV) and Zero-Shot Nouns (ZSN).*

6.3.3 Conclusion of Zero-Shot for Fine-Grained Action Retrieval

In conclusion, by disentangling the caption into its constituent parts of speech and learning an embedding space for each, unseen verbs or nouns can be better retrieved at test time. Using the full caption struggles to capture the semantic information in the same way leading to the model under-performing compared to JPoSE. The substitution of the textual fully connected layer for the gated recurrent unit does lead to some overall improvements however.

Nevertheless, zero-shot retrieval on a large-scale dataset such as EPIC-Kitchens is still a challenging problem. Even with the improvements from using JPoSE, mAP scores still remain low representing an interesting direction for future work.

6.4 Conclusion

Zero-shot tasks, where novel classes are seen during test time which weren't present at training, represent challenging problems in the field of computer vision. Using an open vocabulary for video understanding, by its own nature, leads to long-tail distributions of classes and therefore a higher number of unseen classes.

This chapter has presented two approaches for zero-shot problems for the tasks of action recognition and fine-grained action retrieval. In the case of the former, when using a multi-verb representation, verbs can be thought of as attributes of the actions taking place and therefore novel actions can be explained via the use of previously seen verbs. For fine-grained action retrieval, the JPoSE method was used which disentangles the caption into its constituent parts of speech. By doing this, unseen verbs or nouns can be

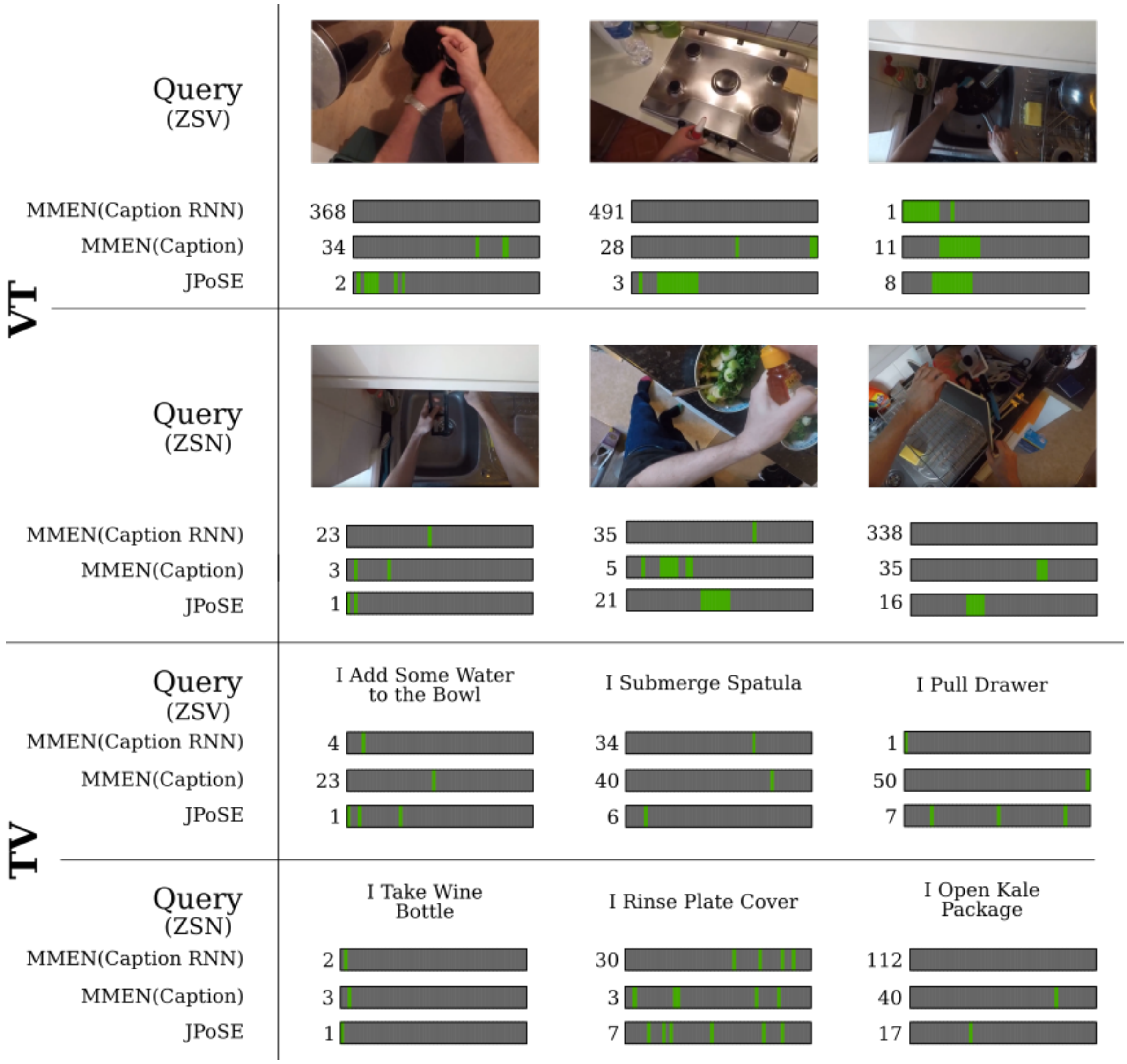


Figure 6.3: *Qualitative zero-shot results on the zero-shot test splits of EPIC-Kitchens. Both video-to-text and text-to-video results are shown for zero-shot verbs (ZSV) and zero-shot nouns(ZSN). For a given query item the rank of the first relevant retrieval is given in addition to the first 50 retrievals showing relevant in green and irrelevant in grey.*

retrieved correctly as the method can focus on parts of the caption that was seen during training.

Using verbs as attributes to describe novel classes proved useful for the task of zero-shot action recognition on the GTEA Gaze+ dataset. Both models trained on multi-verb representations outperformed the model trained with a single-verb representation. The benefit of hard-assigned, multi-verb labels was also highlighted here, with good generalisability being shown across the differing levels of training examples that were provided.

For fine-grained action retrieval, disentangling the caption was demonstrated to be beneficial for the novel action classes, of which many are present for the large number of classes of EPIC-Kitchens. By learning the parts of speech separately, the model wasn't forced to re-learn a new combination of words given a caption from a zero-shot class leading to better results than the baselines which focused on the whole caption.

Chapter 7

Conclusion

Transferring knowledge between language and vision still remains a challenging problem. This thesis serves as a focussed look into verbs and how their relationships can be applied in the field of computer vision, but, as always, more unanswered questions exist in this area. As a whole, the concluding remarks of this thesis are summarised below.

Firstly, verbs represent how we interact with the world, but the relationships between them are highly contextual. These relationships can be hard to discover from corpora or semantic knowledge bases.
Secondly, expanding a closed vocabulary of verbs towards an open vocabulary leads to issues with the standard approach of treating action recognition as a classification problem: a one-vs-all solution does not work here.
Thirdly, verb-only representations can provide large benefits over using the combination of verbs and nouns as action labels, allowing for an object-agnostic representation to be learned.
Finally, disentanglement of captions allows for modelling of the individual parts of an action: the actor, act, and the object(s) being interacted with. This leads to better performance on both within-modal and cross-modal video retrieval tasks.

7.1 Findings and Limitations

In this section, a brief summary of the findings and limitations of each chapter will be presented.

Chapter 3 first explored the issues with using an expanded vocabulary for action recognition. Due to the number of verbs used, many of them had similar meanings and so treating it as a standard classification problem led to cases where either valid verbs were treated as incorrect or one-vs-all approaches were unable to handle the complexity. Additionally, it was found that using WordNet was a hindrance for action recognition — the verb hierarchy within WordNet is too sparse with many synsets having very similar meanings.

Chapter 4 presented the notion of verb-only labelling representations for action recognition where three datasets were annotated with contextually relevant verb labels. Furthermore, different types of verbs — verbs of result and verbs of manner — were investigated along with the relationships between them. The results showed the benefits of a verb-only representation for the tasks of action recognition and action retrieval, notably the soft-assigned labelling method allowing for cross-dataset retrieval. The multi-verb representations required a substantial training effort (30 annotators per video) and therefore is difficult to scale for newer and larger datasets.

Chapter 5 shifted its focus purely onto retrieval, introducing the fine-grained action retrieval task which aims to retrieve semantically relevant items. The Joint Part-of-Speech Embedding (JPoSE) method was also proposed in this chapter which disentangled captions into their constituent parts of speech and created an embedding space for each. This was found to be highly beneficial for EPIC-Kitchens compared to using the entire caption alone. Furthermore, for the general video retrieval task, evaluated on MSR-VTT, caption disentanglement was also shown to be a constructive addition. However, the intra-word relationships are not explored in this method (either within the same Part of Speech or across different ones). Additionally, only RGB and flow features were considered, and the other video representations used in [83] and [76] were not explored.

Finally, Chapter 6 described the tasks of zero-shot recognition and retrieval: challenging tasks where test instances originate from classes not seen during training. The multi-verb representations allowed for unseen classes to be described as combinations of seen verbs, requiring no knowledge of the object being interacted with. The hard assigned multi-verb representation was also found to be generalisable, achieving similar results even with a smaller number of training instances. Similarly to the previous chapter, the decomposition of captions proved fruitful, allowing for JPoSE to outperform caption based approaches when retrieving both videos and captions for unseen combinations of verbs and nouns. Zero-shot retrieval still remains a difficult problem and performance of JPoSE remained low even with the caption disentanglement.

7.2 Directions for Future Work

The work on this thesis has focused on video understanding of actions. Yet, the work in this thesis represents an initial exploration into open vocabulary usage for action recognition and semantic relevancies for action retrieval. As such, there are many exciting avenues for future work within this area. Four interesting topics are highlighted below.

Types of Verbs

Verbs of manner and verbs of result were introduced in chapter 4, however, the relationships between them were only initially explored, leaving considerable scope for future work. One such use case would be imitation learning to teach a system how to perform an action via both the motion and the goal.

To truly explore the types of verbs, it can be expected that both textual and visual understanding would be required. A successful method might employ different features or even modalities to find manner verbs and result verbs. Again, as discussed in chapter 4, the contextual relationships can be hard to discover from semantic knowledge bases or textual corpora so a method that is likely to have to discover this from training data.

Weaker Supervision of Relevancy

Main verbs and supplementary verbs were also introduced within chapter 4 for the multi-verb, verb-only representations. Due to the size of EPIC-Kitchens, it was deemed too costly to annotate the dataset with Soft-Assigned Multi-Verb labels (see section 4.4). Because of this, the notion of supplementary verbs were dropped, with main verbs being related via manual clustering. The manual clustering was later used as a measure of semantic relevancy to train and evaluate the Joint Part-of-Speech Embedding method. For larger datasets, such as HowTo100M [84], this method of relevancy would be unscalable. However, using simply instance based relevancy can lead to issues during both training and evaluation and so weaker forms of relevance represent a viable direction for future work.

Zero-Shot Video Understanding

Chapter 6 saw the application of the multi-verb labels and JPoSE applied to the zero-shot domain. Whilst successful, the performance of these methods still left a lot to be desired, with considerable opportunities of future work still available for both the standard and generalised zero-shot tasks. Methods which are better able to reason about unseen visual elements from textual descriptions are a key starting point.

Further exploration into the generalised zero-shot task is also highlighted here as an interesting direction for future work. This is due to the difficulty of both reasoning about unseen classes as well as determining whether the instance belongs to a seen class or an unseen class. A shared embedding space is important, but a successful method should also focus on reducing the bias of seen classes — potentially through the use of an explicit within-domain classifier.

Joint Modelling of Vision and Language

As discussed in section 2.3.2, recent approaches for performing cross-modal video retrieval have focussed on using different video representations and the visual projection function into the joint space. Other works instead modelled semantic information of the textual embedding function (JPoSE, proposed in chapter 5, falls into the same category). The VideoBERT method, proposed in [128], represents a preliminary work in this direction for classification, retrieval, and captioning. However, this still remains an open area and an interesting direction for future research.

References

[1] Amazon mechanical turk (amt). https://www.mturk.com/. 21, 49

[2] Flickr. https://www.flickr.com/. 43, 44

[3] spaCy. www.spacy.io. 131, 145

[4] YouTube. www.youtube.com. 14, 17

[5] J. Alayrac, P. Bojanowski, N. Agrawal, J. Sivic, I. Laptev, and S. Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *CVPR*, 2016. 59

[6] M. Alka, R. Hovav, and B. Levin. Building verb meanings. *The projection of Arguments*, 1998. 12, 94

[7] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*, 2016. 147

[8] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In *ECCV*, 2014. 43

[9] F. Baradel, N. Neverova, C. Wolf, J. Mille, and G. Mori. Object level visual reasoning in videos. In *ECCV*, 2018. 1, 38, 39

[10] L. Baraldi, F. Paci, G. Serra, L. Benini, and R. Cucchiara. Gesture recognition in ego-centric videos using dense trajectories and hand segmentation. In *CVPR*, 2014. 36

[11] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006. 15

[12] D. A. Behrend. The development of verb concepts: Children's use of verbs to label familiar and novel events. *Child Dev.*, 1990. 12, 94

[13] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman. A short note about kinetics-600. *CoRR*, arXiv:1808.01340, 2018. 21

[14] J. Carreira, E. Noland, C. Hillier, and A. Zisserman. A short note on the kinetics-700 human action dataset. *CoRR*, arXiv:1907.06987, 2019. 21

[15] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *TIST*, 2011. 80

[16] W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *ECCV*, 2016. 168

[17] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, 2011. 20

[18] D. L. Chen and W. B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, 2011. 50

[19] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 2014. 47

[20] E. V. Clark and H. H. Clark. When nouns surface as verbs. *Language*, 1979. 13

[21] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCVW*, 2004. 14, 15, 16, 78

[22] D. Damen, T. Leelasawassuk, O. Haines, A. Calway, and W. Mayol-Cuevas. You-do, I-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video. In *BMVC*, 2014. 1, 3, 25, 30, 58, 59, 60, 61

[23] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018. 1, 14, 25, 41, 129

[24] F. De la Torre, J. Hodgins, J. Montano, S. Valcarcel, R. Forcada, and J. Macey. Guide to the carnegie mellon university multimodal activity (cmu-mmac) database. *Robotics Institute, Carnegie Mellon University*, 2009. 14, 26, 33

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 7, 24, 32, 34

[26] J. Dong, X. Li, C. Xu, S. Ji, and X. Wang. Dual dense encoding for zero-example video retrieval. In *CVPR*, 2019. 54, 55

[27] C. Fang and L. Torresani. Measuring image distances via embedding in a semantic manifold. In *ECCV*. 2012. 70, 73

[28] A. Fathi and J. Rehg. Modeling actions through state changes. In *CVPR*, 2013. 28, 29

[29] A. Fathi, A. Farhadi, and J. Rehg. Understanding egocentric activities. In *ICCV*, 2011. 28, 29

[30] A. Fathi, X. Ren, and J. Rehg. Learning to recognize objects in egocentric activities. In *CVPR*, 2011. 28

[31] A. Fathi, Y. Li, and J. M. Rehg. Learning to recognize daily actions using gaze. In *ECCV*, 2012. 26, 28, 29, 62, 85

[32] C. Feichtenhofer, A. Pinz, and R. Wildes. Spatiotemporal residual networks for video action recognition. In *NeurIPS*, 2016. 41

[33] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016. 1, 19, 20, 34, 39, 40, 112

[34] C. Feichtenhofer, H. Fan, J. Malik, and K. He. Slowfast networks for video recognition. In *ICCV*, 2019. 40

[35] C. J. Fillmore. The grammar of hitting and breaking. *Readings in English Transformational Grammar*, 1967. 12, 94

[36] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell. Compact bilinear pooling. In *CVPR*, 2016. 51

[37] D. Gentner. On relational meaning: The acquisition of verb meaning. *Child Dev.*, 1978. 12, 94

[38] R. Girshick. Fast r-cnn. In *ICCV*, 2015. 45

[39] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *IJCV*, 2014. 44, 45

[40] Y. Gong, L. Wang, M. Hodosh, J. Hockenmaier, and S. Lazebnik. Improving image-sentence embeddings using large weakly annotated photo collections. In *ECCV*, 2014. 44, 45, 146, 153, 157

[41] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. *CoRR*, arXiv:1302.4389, 2013. 51

[42] A. Gordo and D. Larlus. Beyond instance-level image retrieval: Leveraging captions to learn a global visual representation for semantic retrieval. In *CVPR*, 2017. 44

[43] J. Gordo, Albert Almazán, J. Revaud, and D. Larlus. Deep image retrieval: Learning global representations for image search. In *ECCV*, 2016. 43, 44

[44] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thurau, I. Bax, and R. Memisevic. The "Something Something" Video Database for Learning and Evaluating Visual Common Sense. In *ICCV*, 2017. 14, 37, 41

[45] J. Gropen, S. Pinker, M. Hollander, and R. Goldberg. Affectedness and direct objects: The role of lexical semantics in the acquisition of verb argument structure. *Cognition*, 1991. 12, 94

[46] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018. 14, 40

[47] M. Hahn, A. Silva, and J. M. Rehg. Action2vec: A crossmodal embedding approach to action learning. In *BMVC*, 2018. 54, 169, 170

[48] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 44

[49] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *J. Artif. Intell. Res.*, 2013. 44

[50] E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. Ontonotes: the 90% solution. In *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*, 2006. 84

[51] M. J. Huiskes and M. S. Lew. The mir flickr retrieval evaluation. In *ICMR*, 2008. 43

[52] T. Ishihara, K. Kitani, W. Ma, H. Takagi, and C. Asahawa. Recognizing hand-object interactions in wearable camera videos. In *ICIP*, 2015. 30

[53] M. Jain, J. C. van Gemert, T. Mensink, and C. G. Snoek. Objects2action: Classifying and localizing actions without any video example. In *ICCV*, 2015. 24

[54] M. Jain, J. C. Van Gemert, and C. G. Snoek. What do 15,000 object categories tell us about classifying and localizing actions? In *CVPR*, 2015. 24

[55] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, 2008. 43

[56] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, 2007. 16

[57] J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *ROCLING*, 1997. 7

[58] A. Karpathy, A. Joulin, and L. F. Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. In *NeurIPS*, 2014. 44

[59] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 18

[60] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *CoRR*, arXiv:1705.06950, 2017. 1, 21, 145

[61] S. Khamis and L. S. Davis. Walking and talking: A bilinear approach to multi-label action recognition. In *CVPRW*, 2015. 22

[62] A. Kilgarriff and J. Rosenzweig. English senseval: Report and results. In *LREC*, 2000. 133

[63] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. In *TACL*, 2015. 45

[64] B. Klein, G. Lev, G. Sadeh, and L. Wolf. Fisher vectors derived from hybrid gaussian-laplacian mixture models for image annotation. In *CVPR*, 2015. 44, 45

[65] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011. 1, 16, 17

[66] J. Kumar, Q. Li, S. Kyal, E. Bernal, and R. Bala. On-the-fly hand detection training with application in egocentric action recognition. In *CVPRW*, 2015. 27

[67] P. Lade, N. Krishnan, and S. Panchanathan. Task prediction in cooking activities using hierarchical state space markov chain and object based task grouping. In *ISM*, 2010. 33

[68] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 14, 16

[69] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 16

[70] C. Leacock and M. Chodorow. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 1998. 7

[71] M. Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, 1986. 133

[72] Y. Li, Z. Ye, and J. Rehg. Delving into egocentric actions. In *CVPR*, 2015. 27, 28

[73] D. Lin. An information-theoretic definition of similarity. In *ICML*, 1998. 7, 8, 22

[74] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 52

[75] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos in the wild. In *CVPR*, 2009. 16

[76] Y. Liu, S. Albanie, A. Nagrani, and A. Zisserman. Use what you have: Video retrieval using representations from collaborative experts. In *BMVC*, 2019. 53, 181

[77] D. G. Lowe et al. Object recognition from local scale-invariant features. In *ICCV*, 1999. 14

[78] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, 1981. 15

[79] M. Ma, H. Fan, and K. Kitani. Going deeper into first-person activity recognition. In *CVPR*, 2016. 34, 35

[80] F. Mahdisoltani, G. Berger, W. Gharbieh, D. Fleet, and R. Memisevic. On the effectiveness of task granularity for transfer learning. *CoRR*, arXiv:1804.09235, 2018. 37

[81] M. Marszałek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009. 16

[82] T. McCandless and K. Grauman. Object-centric spatio-temporal pyramids for egocentric activity recognition. In *BMVC*, 2013. 32

[83] A. Miech, I. Laptev, and J. Sivic. Learning a text-video embedding from incomplete and heterogeneous data. *CoRR*, arXiv:1804.02516, 2018. 50, 51, 52, 53, 54, 156, 157, 158, 159, 161, 164, 181

[84] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*, 2019. 1, 53, 182

[85] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, preprint arXiv:1301.3781, 2013. 9, 10, 24, 169

[86] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, 2013. 2, 6, 9

[87] G. Miller. Wordnet: a lexical database for english. *CACM*, 1995. 2, 6, 7, 22

[88] N. C. Mithun, J. Li, F. Metze, and A. K. Roy-Chowdhury. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *ICMR*, 2018. 50

[89] M. Moghimi, P. Azagra, L. Montesano, A. Murillo, and S. Belongie. Experiments on an rgb-d wearable vision system for egocentric activity recognition. In *CVPRW*, 2014. 34

[90] D. Moltisanti, M. Wray, W. Mayol-Cuevas, and D. Damen. Trespassing the boundaries: Labeling temporal bounds for object interactions in egocentric video. In *ICCV*, 2017. 105

[91] T. Motwani and R. Mooney. Improving video activity recognition using object recognition and text mining. In *ECAI*, 2012. 1, 22, 23

[92] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006. 43

[93] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 2001. 26, 34

[94] J. Pennington, R. Socher, and C. D. Manning. GloVe: Global vectors for word representation. In *EMNLP*, 2014. 2, 6, 10, 11

[95] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007. 16

[96] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *CVPR*, 2010. 43

[97] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010. 16, 27

[98] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007. 43

[99] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008. 43

[100] L. C. Pickup, Z. Pan, D. Wei, Y. Shih, C. Zhang, A. Zisserman, B. Scholkopf, and W. T. Freeman. Seeing the arrow of time. In *CVPR*, 2014. 39

[101] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *CVPR*, 2012. 30, 31, 32

[102] F. Radenović, G. Tolias, and O. Chum. CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples. In *ECCV*, 2016. 43

[103] K. K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *Mach. Vis. Appl.*, 2013. 16, 17

[104] X. Ren and C. Gu. Figure-ground segmentation improves handled object recognition in egocentric video. In *CVPR*, volume 2, page 6, 2010. 32

[105] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. 1995. 7

[106] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008. 15, 16

[107] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele. A dataset for movie description. In *CVPR*, 2015. 51, 52

[108] M. Rohrbach, A. Rohrbach, M. Regneri, S. Amin, M. Andriluka, M. Pinkal, and B. Schiele. Recognizing fine-grained and composite activities using hand-centric features and script data. *IJCV*, 2016. 14, 37

[109] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 112

[110] M. Ryoo, B. Rothrock, and L. Matthies. Pooled motion features for first-person videos. In *CVPR*, 2015. 33

[111] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *IJCV*, 2013. 24, 78

[112] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *ICPR*, 2004. 16, 41

[113] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *ICLR*, 2013. 78

[114] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, 2014. 33

[115] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2007. 17

[116] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 14, 37

[117] G. A. Sigurdsson, O. Russakovsky, and A. Gupta. What actions are needed for understanding human actions in videos? In *ICCV*, 2017. 92

[118] G. A. Sigurdsson, A. Gupta, C. Schmid, A. Farhadi, and K. Alahari. Actor and observer: Joint modeling of first and third-person videos. In *CVPR*, 2018. 38

[119] G. A. Sigurdsson, A. Gupta, C. Schmid, A. Farhadi, and K. Alahari. Charades-ego: A large-scale dataset of paired third and first person videos. *CoRR*, arXiv:1804.09626, 2018. 37, 38

[120] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014. 19, 21, 34, 36

[121] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, arXiv:1409.1556, 2014. 112

[122] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep fisher networks for large-scale image classification. In *NeurIPS*, 2013. 16

[123] S. Singh, C. Arora, and C. Jawahar. First person action recognition using deep learned descriptors. In *CVPR*, 2016. 35

[124] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *Technical Report CRCV*, 2012. 1, 16, 17

[125] E. Spriggs, F. De La Torre, and M. Hebert. Temporal segmentation and activity classification from first-person sensing. In *CVPRW*, 2009. 26

[126] S. Sudhakaran, S. Escalera, and O. Lanz. Lsta: Long short-term attention for egocentric action recognition. In *CVPR*, 2019. 39

[127] C. Sun, A. Shrivastava, C. Vondrick, K. Murphy, R. Sukthankar, and C. Schmid. Actor-centric relation network. In *ECCV*, 2018. 38

[128] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, 2019. 183

[129] S. Sundaram and W. Mayol-Cuevas. Egocentric visual event classification with location-based priors. In *ISVC*, 2010. 27

[130] E. Taralova, F. De La Torre, and M. Hebert. Source constrained clustering. In *ICCV*, 2011. 1, 14, 26, 33, 62, 85

[131] G. Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of cnn activations. In *ICLR*, 2016. 43

[132] C. Tomasi and T. K. Detection. Tracking of point features. Technical report, Tech. Rep. CMU-CS-91-132, Carnegie Mellon University, 1991. 15

[133] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 1, 15, 24, 78

[134] H. Wang, A. Kläser, C. Schmid, and C. Liu. Action recognition by dense trajectories. In *CVPR*, 2011. 15, 30

[135] L. Wang, Y. Li, and S. Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, 2016. 44, 140

[136] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 40, 145

[137] L. Wang, Y. Li, J. Huang, and S. Lazebnik. Learning two-branch neural networks for image-text matching tasks. *TPAMI*, 2019. 44, 45, 46, 140, 146

[138] D. Wei, J. J. Lim, A. Zisserman, and W. T. Freeman. Learning and using the arrow of time. In *CVPR*, 2018. 39

[139] R. Weischedel, S. Pradhan, L. Ramshaw, M. Palmer, N. Xue, M. Marcus, A. Taylor, C. Greenberg, E. Hovy, R. Belvin, et al. Ontonotes release 2.0. *Linguistic Data Consortium*, 2008. 84

[140] R. Weischedel, M. Palmer, M. Marcus, E. Hovy, S. Pradhan, L. Ramshaw, N. Xue, A. Taylor, J. Kaufman, M. Franchini, et al. Ontonotes 4.0. *Linguistic Data Consortium LDC2011T03*, 2011. 84

[141] R. Weischedel, M. Palmer, M. Marcus, E. Hovy, S. Pradhan, L. Ramshaw, N. Xue, A. Taylor, J. Kaufman, M. Franchini, et al. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium*, 2013. 84

[142] C.-Y. Wu, C. Feichtenhofer, H. Fan, K. He, P. Krahenbuhl, and R. Girshick. Long-term feature banks for detailed video understanding. In *CVPR*, 2019. 40

[143] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *ACL*, 1994. 7, 22, 31, 82

[144] J. Xu, T. Mei, T. Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016. 1, 3, 48, 49, 129, 155

[145] R. Xu, C. Xiong, W. Chen, and J. J. Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI*, 2015. 50

[146] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2014. 43, 44

[147] Y. Yu, H. Ko, J. Choi, and G. Kim. End-to-end concept word detection for video captioning, retrieval, and question answering. In *CVPR*, 2017. 51, 158, 159, 164

[148] Y. Yu, J. Kim, and G. Kim. A joint sequence fusion model for video question answering and retrieval. In *ECCV*, 2018. 51, 158, 159, 164

[149] R. Zellers and Y. Choi. Zero-shot activity recognition with verb attribute induction. *CoRR*, arXiv:1707.09468, 2017. 47

[150] Z. Zhang and V. Saligrama. Zero-shot learning via semantic similarity embedding. In *ICCV*, 2015. 47

[151] Z. Zhang and V. Saligrama. Zero-shot learning via joint latent similarity embedding. In *CVPR*, 2016. 47