UNIVERSITY OF AMSTERDAM

MSc Artificial Intelligence
Master Thesis

Temporal localization of actions in untrimmed videos

by

Juan Buhagiar

11576014

August 12, 2019

36 ECs
March-August 2019

Supervisor:
N Hussein & Dr E Gavves

Assessor:
Dr P.S.M Mettes

Informatics Institute

Abstract:

Action recognition is the process of identifying actions performed by one or more actor/s in a given context based on some observations. Actions come in all shapes and sizes, be it a simple action occurring over a couple of seconds, such as raising your arms, to more complex action occurring over minutes such as vacuuming a room. In this work, we will address the temporal localization problem whereby for a given video we find the start and end times of actions in the video. To achieve this we propose 3 temporal localization networks utilising the ActionVLAD, Non-Local Blocks & VideoGraphs methods that take pre-trained I3D features containing both spatial and temporal information as an input. We compare these methods on the Charades per-frame localization task, with only RGB frames we achieve 14.31%, 11.58 & 13.76% mAP, respectively. We develop and train 3 alternative networks based on the same methods that take in average pooled I3D features in the spatial dimensions, thereby training the networks on temporal information only. ActionVLAD, Non-Local block & VideoGraphs trained on the temporal features obtain 9.373%, 9.408% & 10.73% mAP respectively. We perform in-depth evaluation and analysis of these different models by investigating the Average Precision per class & per noun/verb group used in the generation of the Charades action set. Finally, we visualize the nodes/centroids of the VideoGraphs and ActionVLAD layers to interpret any semantic representations these layers may have learnt.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Problem Definition

Action recognition is the process of identifying actions performed by one or more actor/s in a given context based on some observations. Actions come in all shapes and sizes, be it a simple action occurring over a couple of seconds such as raising your arms to more complex action occurring over minutes such vacuuming a room. Herath et al. [HHP17] define an action as the most elementary human-surrounding interaction. This definition limits actions to humans which is not the case in a broader sense but is sufficient for our work as we will focus on human actions.

Action recognition has been investigated in depth in many occasions [WRB11], [HHP17]. Interest in the area has been shown by works of research even conducted in the 80's [Hog83]. Works in this area have mainly been applied to visual observations such as video but not exclusively, there are works which use other observations such as audio [OMK+14]. There are different subsets of the action recognition problem, namely, Temporal localization, Spatio-temporal localization and Spatio-temporal localization and tracking. In this work, we will focus on Temporal Localization of actions in untrimmed videos.

The Temporal Localization task we are addressing involves an untrimmed video in which actions occur by individuals or objects. The problem is that we do not know in which time points these actions are occurring. Therefore, a system has to be developed that parses the video and assigns an action, several actions or no actions to each time points. Figure 1 shows a visualization of this problem whereby, a sequence of video frames are assigned different action classes at several time points.

Applications of temporal localization are diverse, with many applications having deep economical & societal impact leading to a high interest in this research area. With the explosion of research in Artificial Intelligence many have started to speculate and imagine a future whereby robots become a more frequent occurrence in our life. What makes human communication so unique is not only our ability to speak and write but also to understand others based on their behaviour and react to that. Thereby we can imagine how action recognition can be a basis for such works in improving robot perception and open up further possible applications. Another interesting application of action recognition is in the area of smart cities. Smart cities, when implemented well can improve many problems people experience in cities. Problems that can be addressed with action recognition are numerous for example identifying littering behaviour, recognition of violent behaviour and

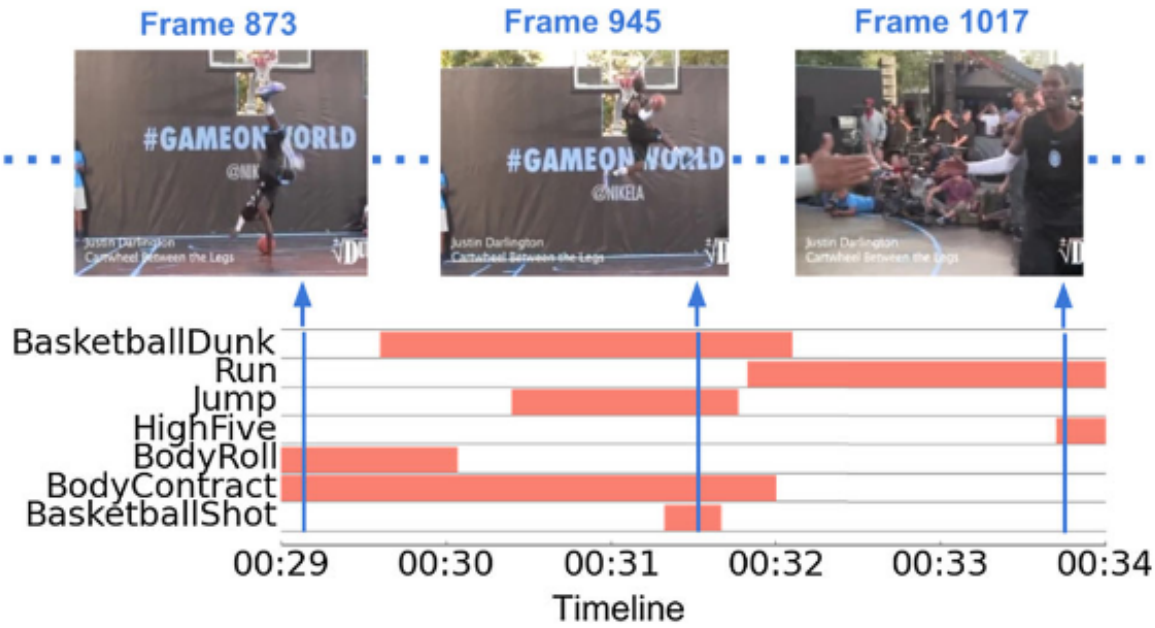

Figure 1: The Temporal Localization Problem visualized, above we can see a sequence of video frames which are assigned an action, multiple actions or no actions to each time point. Reproduced from [YRJ+18].

many others. Other application areas include surveillance, healthcare, video retrieval etc.

The focus of this research as mentioned above is to solve the temporal localization problem by identifying and implementing the most effective methods of temporal modelling to localize actions by humans in videos. As well as to analyse and understand the behaviour of the modelling techniques used.

1.2 Motivation

In recent years, we have seen many technological advancements that have enabled us to perform research and work in fields that were not feasible before. There are many examples of this, activity recognition is no exception as the computational complexity of current methods required large computation and memory requirements that were not possible in the past. Withal, a large number of successful works in the domain of computer vision and deep learning have been published which enables us to build on top of this work. With the ever-increasing developments, more and more challenges/datasets become publicly available.

Apart from the feasibility of such research, Artificial Intelligence has become a focus

point in many industries and even countries with countless forward-looking ideas on how AI can be applied to tackle very different problems. However, for these solutions to become reality there are several obstacles we need to overcome. Computer Vision is one of the domains in which many modern applications rely on. We think that with the improvement of Action Recognition research, many new applications will start to emerge and many others would improve.

1.3 Scope

In this work, we will describe and test different approaches to temporally localize action in untrimmed videos. A dataset of videos where human individuals perform actions will be used to temporally localize these actions. The dataset will contain dense labelling with a frame-level annotation rather than video-level. We will focus our efforts on the temporal modelling aspect and not in designing an end-to-end temporal localization network. Therefore, for an effective execution of this project, we will use pre-trained 3D CNN to extract features from the videos.

1.4 Aims & Objectives

In this project, we aim to tackle the Activity Recognition challenge of the Charades dataset. We will develop a pipeline and investigate order-less temporal modelling approaches such as Self-Attention, ActionVLAD & VideoGraphs. Specifically, we aim to answer the following questions:

- Are ActionVLAD, Self-Attention & VideoGraph methods amenable to temporal action localization in untrimmed videos?
- Specifically, can we use these methods as part of a model to temporally localize actions in videos on a per-frame level rather than video-level?
- Given that we extract a temporal feature set and a Spatio-temporal feature set from a video. Do models trained on the temporal feature set generalize differently than the models trained on the Spatio-temporal feature set?

1.5 Report Layout

The Background and Related works sections can be found in Section 2 & 3, respectively. Chapter 4 includes the Methods section. The evaluation & results of the system are

described in Chapter 5 and the discussion is noted in Chapter 6. The future works can be found in Chapter 7 & the conclusion is discussed in Chapter 8.

2 Background

In this section, we describe techniques that are referenced throughout the research. These techniques are the basic concepts used in the implementation of our research.

2.1 3D Convolutional Neural Networks

Deep learning methods have become significantly better than most hand-crafted methods. These methods have achieved competitive performance in many challenges in domains such as Natural Language Processing, Audio classification, denoising, human-machine interaction and much more. Namely, Convolution Neural Networks (CNNs) have been in use in the computer vision domain for countless challenges. CNNs are a type of deep learning model that learn trainable filters and utilise pooling operations over local neighbourhoods on raw image inputs. It has been demonstrated that CNNs can to be invariant on some variations such as pose, lighting etc.

CNNs have been mainly utilised as 2D CNNs such that they are applied to individual image frames. To be able to extend these works to videos, Ning et al. [NDL+05] applied 2D CNNs to frame-level images from videos, however, this approach misses out on motion information present in contiguous frames of a video. To be able to capture spatial information in the images and also temporal information from multiple contiguous frames Ji et al. [JXYY12] propose a 3D Convolutional Neural Network, where the conventional 2D convolutions in CNNs are replaced with 3D substitutes.

3D convolutions are applied to both the spatial and temporal dimensions. This is achieved by performing multiple convolutions at the same location using a 3D kernels to a number of stacked contiguous frames. By this formulation, feature maps in the convolution layer are linked to multiple contiguous frames, thereby capturing motion information. Mathematically, the value of the (x,y,z) on the jth feature map of the ith layer is defined as:

$$v_{ij}^{xyz} = tanh(b_{ij} + \sum_{m} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{ijm}^{pgr} v_{(i-1)m}^{(x+p)(y+q)(z+r)}) \quad (1)$$

where R_i the size of the 3D kernel along the temporal dimension, w_{ijm}^{pqr} is the $(p, q, r)th$

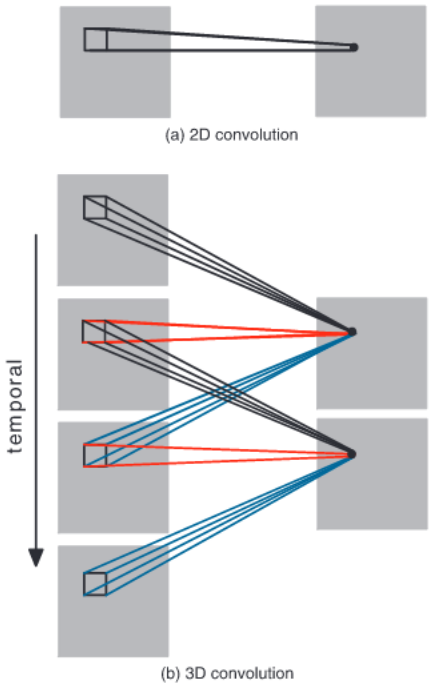

Figure 2: Comparison of (a) 2D and (b) 3D convolutions. Reproduced from [JXYY12]

value of the kernel connected to the *mth* feature map in the previous layer. A comparison of 2D and 3D convolutions is given in Fig. 2.

2.1.1 Two-Stream Inflated 3D ConvNets (I3D)

Prior to the works presented by Carreira et al. [CZ17], most 3D CNNs were pretty shallow, this was mainly due to the higher dimensionality of parameterization with deeper networks and the lack of labelled video data. Carreira et al. observed that there are countless works on image classification that have achieved state-of-the-art results such as Inception, VGG-16 & ResNet. The authors show that these networks can be inflated into Spatio-temporal feature extractors. This is done by inflating all filters and pooling kernels in the networks, which results in an additional dimension for temporal information. Specifically, most filters have a dimension of a square NxN which will be inflated to a cube $NxNxN$. Apart from the architecture, the authors looked at bootstrapping the pre-trained weights from 3D networks to 2D. To achieve this, 2D weights are repeated N times along the temporal dimension and are rescaled and divided by N. They show that a two-stream network is ideal, therefore they train an RGB model and an Optical flow model. The final models trained are an inflated Inception V1 network with bootstrapped weights from training on

ImageNet. Finally, they train the inflated network on the Kinetics dataset [KCS+17].

2.2 Non-Local Approaches

While CNNs consider local neighbourhoods to compute the convolutions, non-local approaches compute responses as a weighted sum of features at all locations.

2.2.1 Self-Attention & Non-local Networks

Self-attention [WGGH18] has been extensively used in NLP, specifically in Natural Language Understanding. Self-attention computes responses at a position in a sequence by attending to all positions and taking a weighted average in an embedding space. The authors consider self-attention as a form of non-local mean which allows us to utilise the concept of self-attention in computer vision. Formally a non-local operation is defined as:

$$y_i = \frac{1}{C(x)} \sum_{\forall j} f(x_i, x_j) g(x_j) \tag{2}$$

Where i is the index of an output position to be computed and j enumerates all possible positions. x is the input feature and y is the output feature of the same size as x. A pairwise function f computes a scalar representing a relationship between i and all j. The function g computes a representation of the input signal at the position j. Normalization is done with the factor C(x).

There are different versions of the f & g functions, the authors consider g as a form of linear embedding, where g is implemented as a convolution over space or spacetime. They present different functions for the pairwise function f, namely, a Gaussian function, an Embedded Gaussian function, Dot-product and Concatenation. To be able to incorporate the non-local mean operation in any network, the authors define a non-local block as follows:

$$z_i = W_z y_i + x_i \tag{3}$$

where y_i is defined by Eq. 2 and x_i is the residual connection which allows for the non-local block to be inserted into any pre-trained model without breaking the original behaviour. The non-local block is visualized in Figure 3.

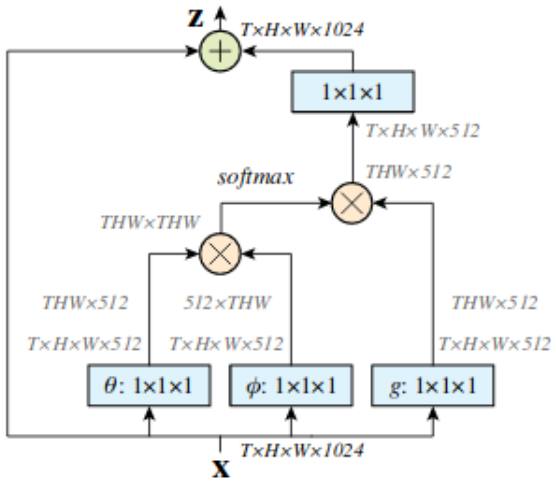

Figure 3: Spacetime Embedded Gaussian non-local block visualization, reproduced from [WGGH18]

2.2.2 VideoGraph

Hussein et al. [HGS19] propose a method for modelling minute-long temporal dependencies called VideoGraph. The idea is to learn an undirected graph representation for human activities. These graph representations such as nodes and edges are learnt entirely from the video dataset. Graph nodes N are representation of a dominant latent short-range concept called a unit action. Unit actions can be though of as building block of human activities.

To represent unit actions or graph nodes, the authors propose to learn a latent features $Y, Y = \{y_j | j = 1, 2, ..., N\}, Y \in \mathbb{R}^{NxC}$ which are vector representations of graph nodes N. Then, for each feature x_i a correlation is calculated with the vector representations Y of the nodes. To achieve this, they propose the node attention block, which takes an input feature x_i as input and node features Y. Formally the node attention block is defined as follows:

$$\hat{Y} = w * Y + b \tag{4}$$

$$\alpha = \sigma(x_i * \hat{Y}_i^T) \tag{5}$$

$$\begin{aligned} Z_i &= \alpha \odot \hat{Y} \\ &= \alpha_j \odot y_j, j = 1, 2, ..., N \end{aligned} \tag{6}$$

Equation 4 is a one layer MLP transformation that allows the node learnable and better suited for the dataset presented. Activations are collected in Equation 5 by applying an activation function on the dot product similarities between x_i & $\hat{Y}$ such that non-linearity is applied. Finally, all the nodes $\hat{Y}$ are multiplied with the activation values α to obtain a representation, Z_i, of segment s_i in the same feature space as the nodes. A representation of the node attention block is depicted in Figure 4a.

Furthermore, the authors propose a way to learn graph edge, ε to obtain a full graph structure. To achieve this they propose a new graph embedding layer, depicted in figure 4b. This layer is designed to extract information from two types of relationships, firstly, from the temporal aspect of unit actions and secondly, between unit-actions themselves. To learn both of these relationships they utilise a 1D Convolutions over time and the nodes. For learning the relationship between nodes, given the graph nodes learnt $\{Z_i, ..., Z_t\}$, a node-wise 1D convolution is applied, that is, a convolution over the elements in each Z_i representation. As for learning a temporal relationship between nodes, the convolution is applied to the temporal dimension only.

Both of these convolutional layers learn different graph edges for each channel, therefore, a 2D convolution is applied to model cross-channel correlations in each node feature $z_i j$. Subsequently, the resulting graph edges are passed through BatchNormalization & Relu Layers. Finally, the full graph representation Z is down-sampled over time and node-dimension using a MaxPooling operation with kernel size and stride of 3.

2.3 VLAD, NetVLAD & ActionVLAD

Vector of Locally Aggregated Descriptors (VLAD) [JDSP10] is a popular aggregation of feature descriptor created as a simplification of the Fisher kernel. VLAD stores information about the statistics of local descriptors aggregated over the image. In contracts to the bag-of-words method which keeps a count of visual words, VLAD stores a sum of the vector representation of the difference between the descriptor and cluster centroids for each visual word, called residuals. Formally the VLAD descriptor is defined as follows:

$$V(j,k) = \sum_{i=1}^{N} a_k(x_i)(x_i(j) - c_k(j)) \tag{7}$$

where $x_i(j)$ and $c_k(j)$ are the j-th dimensions of the i-th descriptor and k-th cluster centre, respectively. $a_k(x_i)$ denotes the membership of the descriptor x_i to cluster c_k.

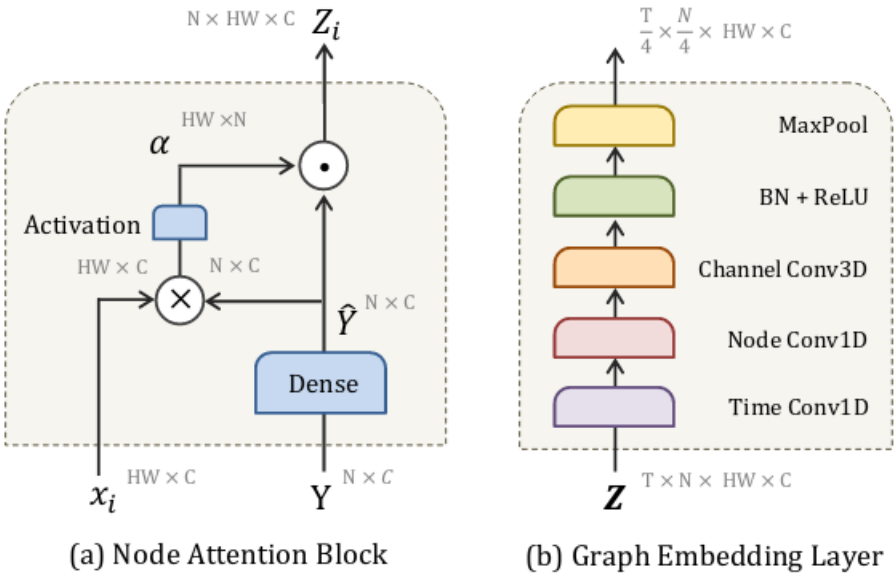

Figure 4: The node attention block reproduced from [HGS19]

Arandjelovic et at. [AGT+16] propose NetVLAD to mimic the behaviour of VLAD in a CNN layer to represent an image. The network they propose can be trained as an end-to-end network. To achieve this, the VLAD layer must be differentiable, the authors recognise that the hard assignment $a_k(x_i)$ of descriptors x_i to cluster centres c_k is not differentiable. Therefore, they propose a soft-assignment of the descriptors to multiple clusters, formally:

$$\bar{a}_k(x_i) = \frac{e^{-\alpha||x_i - c_k||^2}}{\sum_{k'} e^{-\alpha||x_i - c'_k||^2}} \tag{8}$$

$$= \frac{e^{w_k^T x_i + b_k}}{\sum_{k'} e^{w_{k'}^T x_i + b_{k'}}} \tag{9}$$

Equation 9 is a simplified version of Equation 8 by expanding the squares. Substituting the soft-assigment back into the VLAD descriptor in Equation 7 we obtain the following:

$$V(j,k) = \sum_{i=1}^{N} \frac{e^{w_k^T x_i + b_k}}{\sum_{k'} e^{w_{k'}^T x_i + b_{k'}}} (x_i(j) - c_k(j)) \tag{10}$$

The NetVLAD layer as seen in 10 has three sets of learnable parameters w_k, b_k & c_k in contrast to VLAD which only learnt the clusters c_k. This gives NetVLAD greater flexibility, ability to learn in an end-to-end manner in a CNN & aggregates first-order statistics of

residuals much like the original VLAD

Girdhar et al. [GRG+17] propose an extension of the NETVLAD layer for Spatio-temporal aggregation. They propose a very similar ActionVLAD layer which has an extra aggregation over time, formally, the VLAD descriptor is calculated as follows:

$$V(j,k) = \sum_{t=1}^{T} \sum_{i=1}^{N} \frac{e^{w_k^T x_{it} + b_k}}{\sum_{k'} e^{w_{k'}^T x_{it} + b_{k'}}} (x_{it}(j) - c_k(j)) \tag{11}$$

3 Related Work

3.1 Datasets

In this section we will discuss varies datasets which have been used for the temporal localization problem in untrimmed videos. An important factor in action detection is the length and complexity of actions, some dataset contain actions which are relatively short such as the Kinetics [KCS+17] & AVA [GSV+17] datasets. Others datasets contain actions which occur over a longer period of time such as Charades [SVW+16], & MultiTHUMOS [YRJ+18].

Atomic Visual Actions (AVA) dataset is a video dataset containing 80 atomic visual actions which are densely labelled across 430 15-minute videos. This leads to 1.58 million labels with multiple labels occurring per person frequently. The dataset is focused on defining atomic actions rather than complex composite actions. The latest release of the Kinetics dataset, Kinetics-700, is relatively larger dataset containing high-quality videos from YouTube. These videos include a range of human-focused actions spread over approximately 650K videos covering 700 human action classes. Each class contains at least 600 video clips and last around 10 seconds.

MultiTHUMOS is an extension of the THUMOS dataset, containing 30 hours of video across 400 videos with dense, multilabel, frame-level action annotations of 65 classes. In total, the dataset contains 38.5K annotations with a density of around 1.5 actions per frame & around 10.5 actions per video. The Charades dataset, named after the popular American game, is a publicly available dataset containing 9848 videos of everyday activities that occur at home. The dataset is collected in 15 types of indoor scenes, involves interactions with 46 object classes and has a vocabulary of 30 verbs leading to 157 action classes. It has 66,500 temporally localized actions, 12.8 seconds long on average, recorded by 267 people in three continents. Figure 5 depicts sample frames of actions from the dataset.

3.2 Initial works, Holistic Representations & Local descriptors

Initial work in action recognition has been done on 3D models to describe actions. Although methods such as WALKER [Hog83] have proven to be effective, 3D models are difficult and expensive to construct. Subsequently, research followed in two directions for alternate representations of actions namely Holistic & Local representations. With the success of Deep Learning, most research nowadays is focused on Deep Neural Networks.

Approaches using Holistic Representation rely on a global representation of human body structure, movement and shape. Most interestingly, Bobick et al. [BD01] introduced a new representation for human movement by utilising temporal templates. Two templates were introduced the Motion Energy Image (MEI) and the Motion History Image (MHI). MEI describes where motion happens while MHI shows how the MEI is moving. These templates have been used in many other works as they contain useful information about the context of the videos. Tian et al. [TCLZ12] proposed a Hierarchy Filtered Motion method that filters out moving and cluttered backgrounds by utilising the gradient of the MHI template.

These techniques [HS$^+$88], [BD01], [YS05] have produced some interesting works, which at the time were considered state-of-the-art. However, researchers [DRCB05], [MHS09] often found that the holistic approaches were far too rigid to handle the many possible variations of actions.

Local descriptors became popular and soon outperformed Holistic Approaches. Local approaches, as with images, require keypoint detection, keypoint local description and aggregation of local descriptors. Initial work in local descriptors involved extending 2D keypoint detection, local descriptors and aggregation to 3D ones [Lap05], [WTVG08].

Due to an issue with the spareness of keypoints, works started to appear that disintegrate spatial locations from the temporal one [DRCB05]. This is based on the intuition that actions do not have to occur on the same spatial location for its duration, this fueled works with trajectories [MPK09], [MHS09] as opposed to cuboid based works [DRCB05], [Lap05].

3.3 3D CNNs

As mention in Section 2.1, a large number of works in CNNs in recent years has enabled the success of many research works in the area of computer vision. Ji et al.[JXYY12] introduced 3D convolutions which enabled CNNs to perform convolution in the spatial and

temporal domain by using 3D kernels. The authors show that their method outperforms other methods on the TRECVID dataset.

Some of the downsides of 3D CNNs are that they have a rigid temporal structure as they accept a pre-defined number of frames as an input. Moreover, they require a large number of parameters to be trained, much greater than that of 2D CNNs.

Kong et al. [TBF+14] extend 3D CNNs to a more modern architecture named C3D, containing 5 convolutional layers, 5 max-pooling layers, 2 fully connected layers and a softmax layer. The network was extensively tested on 4 different tasks and 6 different benchmarks by using features from the network and a simple linear classifier. Specifically, on action recognition, they obtain an accuracy of 85.2% on both UCF101 & Sport1M.

There are many ways to model temporal information, in general, Ng et al. [YHNHV+15] investigate different convolutional temporal feature pooling methods and found that max-pooling operations are preferable.

Karpathy et al. [KTS+14] introduce the concept of slow fusion to increase the temporal support of a CNN. Slow fusion, a convolutional network accepts several consecutive frames of a video and processes feed them to the same layers to produce activations across the temporal domain. These activations are then fed to fully connected layers to obtain features. The authors also propose a multi-resolution CNN consisting of two identical networks accepting different resolution images. The multi-resolution CNN can learn faster with no cost to the accuracy. The authors show significant performance improvements compared to the UCF-101 baseline 63.3% accuracy up from 43.9%

Many works followed that use 3D CNNs for action detection [SJYS15], [DPVG16], [MSPGiN16]. By extending kernels to 3D learnable parameters in CNNs increased significantly, Sun et al. suggested a factorization of a 3D kernel into a combination of a 2D and 1D kernel [SJYS15] that has comparable performance as the 3D one but contains significantly less learnable parameters.

Girdhar et al. [GRG+17] introduce a new video representation for action classification that aggregates local convolutional features across the entire Spatio-temporal extent of the video inspired by the VLAD descriptor named ActionVLAD. They investigate into which strategies for pooling across space and time perform best as well as benchmarking their representation on different datasets. They achieve 93.6% & 69.8% accuracy on UCF101 & HMDB51, respectively with a two-stream RGB & optical flow network for video-level classification. With an RGB network only they achieve 21% mAP on the Charades dataset.

Carreira et al. [CZ17] as described in section 2.1.1 introduce a new two-stream inflated

3D CNN that is based on 2D CNN inflation. Utilising only the RGB stream, they obtain state-of-the-art activity classification on video-level for the UCF101, HMDB51 & Kinetics datasets with accuracy rates of 84.5%, 49.8% & 71.1%, respectively.

3.4 Non-Local

The recent success of the self-attention in the area of NLP [VSP+17] has inspired work in computer vision with non-local operations. While CNNs and RNNs consider one local neighbourhood, Non-local operations are ideal to capture long-range dependencies. Wang et al. [WGGH18] propose a new building block for Neural networks which is a spacetime equivalent of non-local mean operation. They achieve a 77.7% top-1 accuracy and 93.3% top-5 accuracy on the Kinects dataset.

Wu et al. [WFF+18] propose a pipeline for long-term feature banks using non-local operations. These long term-feature banks are support features extracted from the entire video/clip. The long term feature banks can be looked at as a 'memory' of what has happened throughout the whole input video. These feature banks can be part of any state-of-the-art video model that would only view short clips otherwise. Their experiments show that incorporating 3D CNNs with a long-term feature bank gives state-of-the-art results AVA, EPIC-Kitchens and Charades for action recognition.

3.5 Temporal Localization

Gaidon et al. [GHS13], [GHS11] introduced the problem of temporal localization in untrimmed videos focusing on a limited action set such as 'drinking and smoking'. As more large scale untrimmed video datasets became available with fine-grained actions more works started to be conducted.

To better exploit temporal information some works have focused there works on RCNNs or specifically LSTMs. Due to the recurrent nature of RNNs, they are ideal for time-series modelling. Baccouche et al. [BMW+11] propose a method for extracting features with 3D CNNs and classifying actions by feeding these features to an LSTM. Donahue et al. [DAHG+15] propose a recurrent convolutional architecture which is end-to-end trainable and suitable for action recognition. Li et al. [LQY+16] consider different granularities of a video such as a single frame, a clip or the entire video. These granularities are modelled with CNNs and also fed to an LSTM to leverage temporal queues.

Yeung et al. [YRJ+18] introduce the MultiTHUMOS dataset with multi-label anno-

tations for every frame in the THUMOS dataset. The authors also propose an LSTM network to model multiple input and output connections to predict frame-level annotations for multiple classes. They achieve an mAP of 41.3% and 29.7% on THUMOS and MultiTHUMOS, respectively.

Shau et al. [SCZ+17] propose a novel network called Convolutional-De-Convolutional (CDC) network, that places CDC filters on top of a 3D CNN. The authors suggest that most prior works perform temporal localization at segments level. The CDC filters perform temporal upsampling and spatial downsampling operations simultaneously to predict actions at frame-level granularity. Their model beat the current state-of-the-art on per-frame labelling on THUMOS'14 with an mAP of 44.4%.

Piergiovanni et al. [PR18] propose a novel way to learn latent super-events from activity datasets. The authors define super-events as a set of multiple events occurring together in videos with a particular temporal organization. To be able to capture how events are temporally related in videos, they propose a temporal structure filter which can be included in a deep learning model and trained end-to-end. The authors utilise the I3D network to extract features from video segments. Their best model achieves mAP of 36.4%, 19.41% & 8.3% on MultiTHUMOS, Charades & AVA dataset, respectively.

4 Method

In this section, we will discuss the approach taken to temporally localize actions in untrimmed videos. Our approach utilises pre-trained 3D CNNs & temporal modelling networks.

4.1 Dataset

We have chosen to use the Charades dataset for our experiments 3.1 which contains 9848 videos of daily activities with an average action time of 30.1 seconds. The dataset is collected in 15 different indoor scenes involving a vocabulary made up of 46 object classes and 30 verb classes to generate 157 action classes. The dataset contains 66.5K temporal instances recorded by 267 individuals in three continents.

The dataset was created by using the popular Amazon framework Amazon Mechanical Turk (AMT), whereby individuals submitted videos based on scripts generated by the authors. To generate the scripts, the authors analysing the TF & TF-IDF of 549 popular movies and created the 30 verb classes and 46 object classes from the most occurring verbs and nouns. Workers on AWT were presented with a scene, 5 randomly selected nouns

and 5 selected verb actions. The workers generated scripts using 2 nouns and 2 verbs of their choice about realistic and commonplace activities in their home. These scripts were presented to other works which acted out the script and recorded themselves doing so. The videos were then given to other workers to describe what is happening in the video while others were assigned to temporally annotate the videos.

Figure 5 shows some sample actions classes and a respective frame sample of that action. Each row of the figure is extracted from the same video, which could have multiple action occurring in a single frame. Hence, some frame samples look very similar yet are assigned a different annotation, in reality, it might be that there are multiple annotations for that frame.

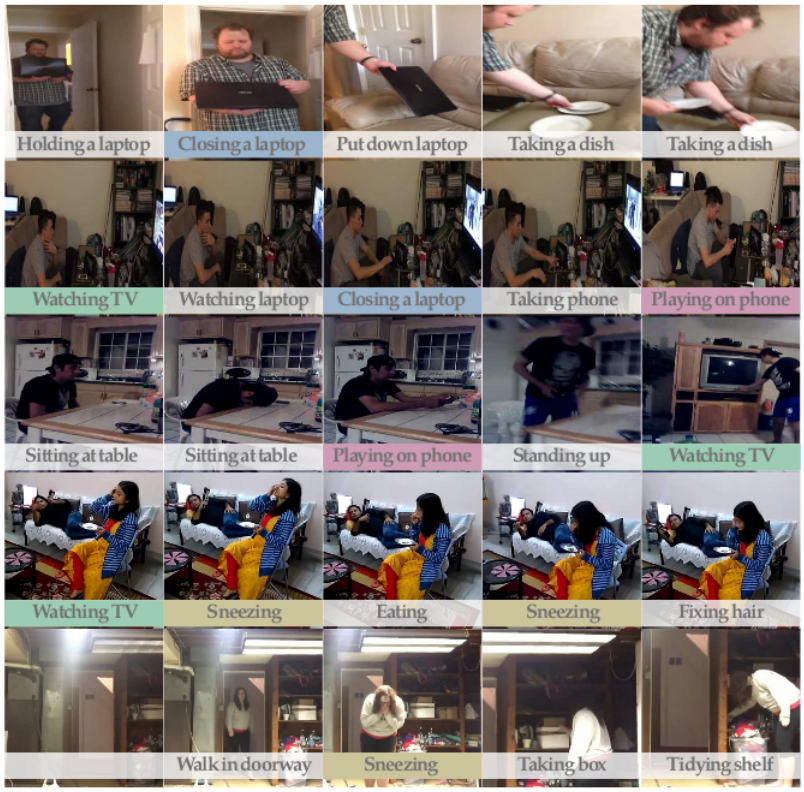

Figure 5: Charades Dataset samples, reproduced from [SVW+16]

4.2 Feature Extraction

The process of temporally localization actions starts off with our video files, V. We sample the video files @ 24 fps generating thousands of frames from the dataset. We extract

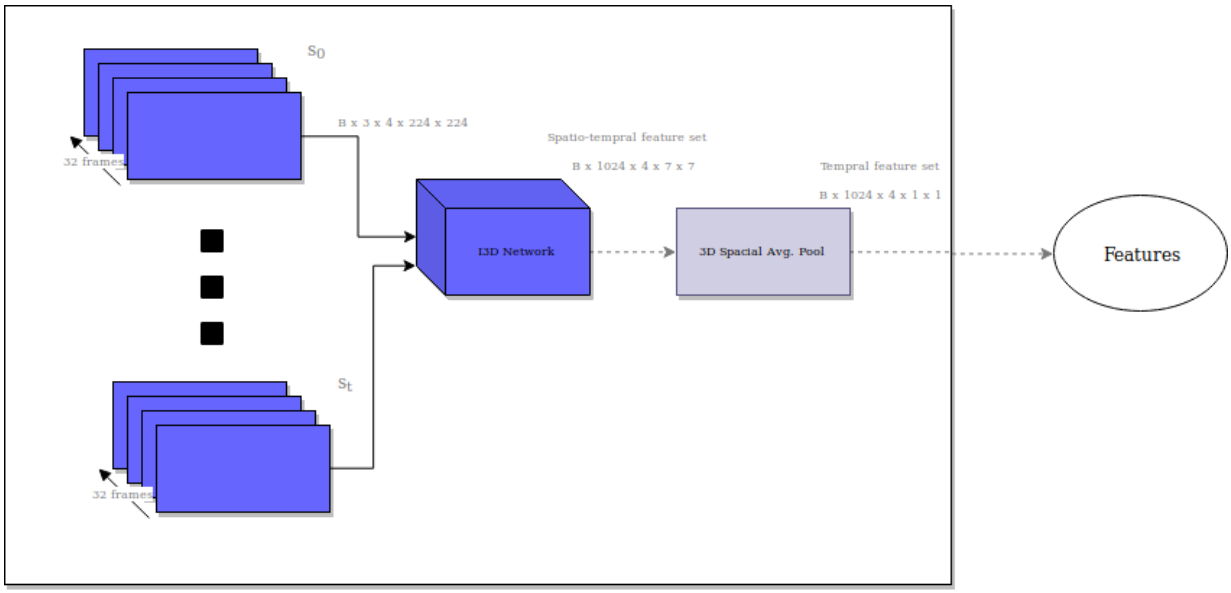

Figure 6: **Feature extraction pipeline:** Video files are split into 32 contiguous frame segments and are fed to the I3D networks saving the **Mixed5c features**. This leads to the Spatio-temporal feature set. The 3D spacial Average Pooling is only applied to obtain the temporal feature set.

features using the RGB stream of the I3D network described in section 2.1.1, which was initially trained on the Kinetics dataset and fine-tuned in the Charades dataset.

The extracted frames are of different resolutions, therefore, a centre crop is applied to the frames at a resolution of 224. Features are extracted by first generating segments s_i for every 32 consecutive frames. Each segment s_i is feed into the I3D network and the 'Mixed_5c' features F are generated. Formally, for each video v_i we obtain segments of 32 consecutive frames represented as $x_i \in \mathbb{R}^{Bx3x32x224x224}$. The resulting features will be a representation of 4 timesteps, due to the temporal grouping of 8 frames by the I3D network, in the form of $f_i \in \mathbb{R}^{BxCx4x7x7}$. Where C is the channel size & B is the batch size. For all of our experiments, the channel size was set at 1024. The final feature set is the Spatio-temporal features, to further understand the temporal models we create a set of temporal feature set. These are obtained by passing the Spatio-temporal features through a 3D Average Pooling layer over the spatial domain. This leads to a feature set in the form of $f_i \in \mathbb{R}^{BxCx4x1x1}$. Figure 6 depicts the feature extraction process.

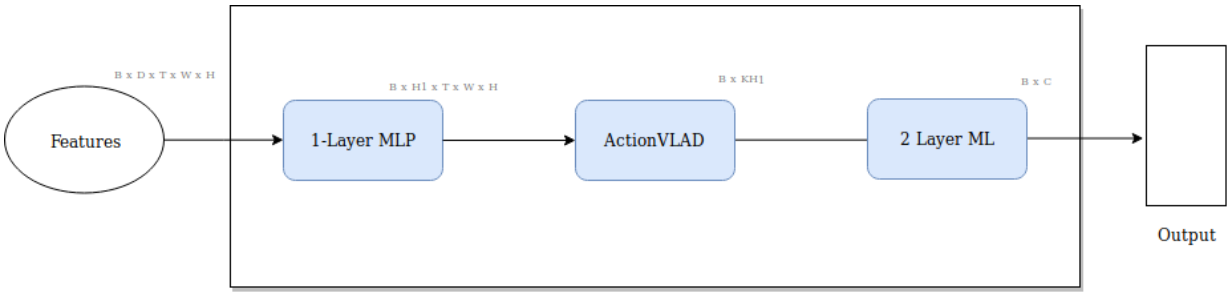

Figure 7: Network A: **ActionVLAD** Temporal Localization Network.

4.3 Temporal Localization

In this step, we propose our temporal localization networks utilising different orderless statistical approaches. The networks we propose are fed features vectors F and output frame-level predictions of action classes.

4.3.1 Network A: Utilising ActionVLAD

ActionVLAD has shown to be an effective aggregation technique for Spatio-temporal video representation. For our first model, we propose a Temporal localization network that utilises the ActionVLAD layer proposed in [GRG+17]. The network is set up with a 1-Layer MLP & an ActionVLAD layer to learn a Spatio-temporal aggregation for temporal modelling. The representation is then fed to the 2-layer MLP which is used as a classifier for action detection.

Specifically, we start with a feature vector, $f_i \in \mathbb{R}^{BxDxTxWxH}$, representing a segment s_i. We transform the features using a one hidden layer MLP with a dropout, batch normalization & a leaky relu. The transformed feature $x_i \in \mathbb{R}^{BxH1xTxWxH}$ is fed into the ActionVLAD layer described in section 2.3. The soft-assignment over the clusters is calculated as described in equation 8, leading to activations $\hat{\alpha} \in \mathbb{R}^{BxTWHxClusterSize}$. We then use equation 10 to calculate the VLAD descriptor with soft-assignment, $V_h \in \mathbb{R}^{BxH1xClusterSize}$. The VLAD descriptor is then normalized and reshaped into a 1 dimensional vector representation $V \in \mathbb{R}^{BxClusterSizeH1}$. Finally, to classify the features to action classes we pass the normalized VLAD vector through a two hidden layer MLP with Dropout layers, Batch Normalization & Leaky Relu. Figure 7 depicts the setup of network A.

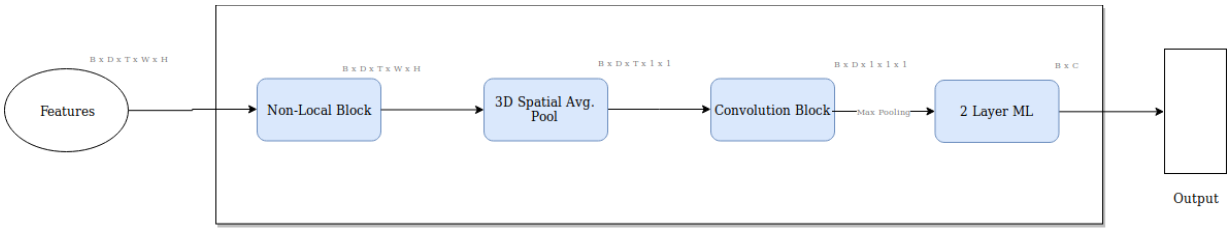

Figure 8: Network B: **Non-Local Block** Temporal Localization Network.

4.3.2 Network B: Utilising the Non-Local Block

Non-Local & self-attention based operators have shown impressive results in modelling long-term dependencies in different tasks. We propose our second model that utilises the Non-local Block propose by [WGGH18]. The network first uses the non-local block to obtain a response over both space and time, this is then fed into a 3D average pooling layer and a convolutional layer to aggregate the features over space and time. Finally, a 2-layer MLP is used as a classifier for the action detection task.

Similar to before we start with a feature $f_i \in \mathbb{R}^{BxDxTxWxH}$ representing a segment s_i. The features are initially fed into a 3D non-local block as described in section 2.2.1, producing features of the same size. Unlike the other methods, the non-local block outputs features of the same size. Therefore, we use average pooling over the spatial dimensions with a kernel of [1,7,7]. For the temporal dimension, we use a convolutional layer with a kernel of [4,1,1] followed by a Batch Normalization & Leaky Relu layers which we call the Convolution Block. Finally, we pass the features to a hidden two-layer MLP with Dropout layers, Batch Normalization & Leaky Relu. Figure 8 depicts the setup of network B.

4.3.3 Network C: Utilising VideoGraphs

Our last model utilises a recently proposed modelling technique that is able to learn minute-long dependencies & the temporal structure of actions called VideoGraph [HGS19]. The network is set up very similar to the paper proposed, we learn a graph representation by passing the features & randomly initialised centroids through a Node Attention layer & a Graph embedding layer. The Graph representation is Max Pooled over space and time and is fed to a 2-Layer MLP classifier to predict action classes.

For node attention we start with a feature $f_i \in \mathbb{R}^{BxDxTxWxH}$ representing a segment s_i & a randomly initialised set of centroids $Y \in \mathbb{R}^{NxC}$. The features & centroids are fed into the node-attention block as described in section 2.2.2 to obtain a node representation Z_i. To achieve this we have to learn the graph edges ε. Graph edges, ε, we feed the node

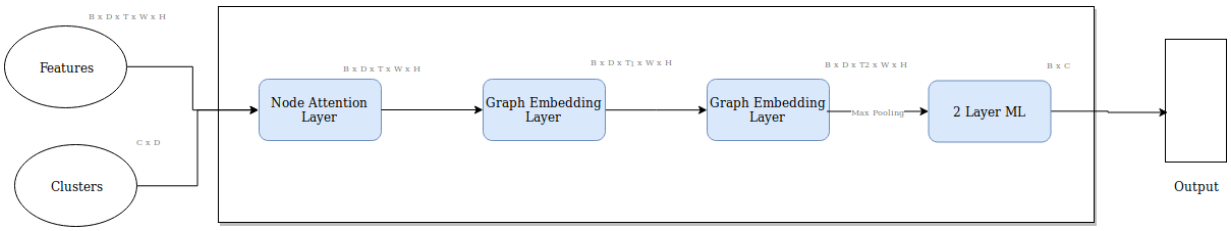

Figure 9: Network C: **VideoGraph** Temporal localization network

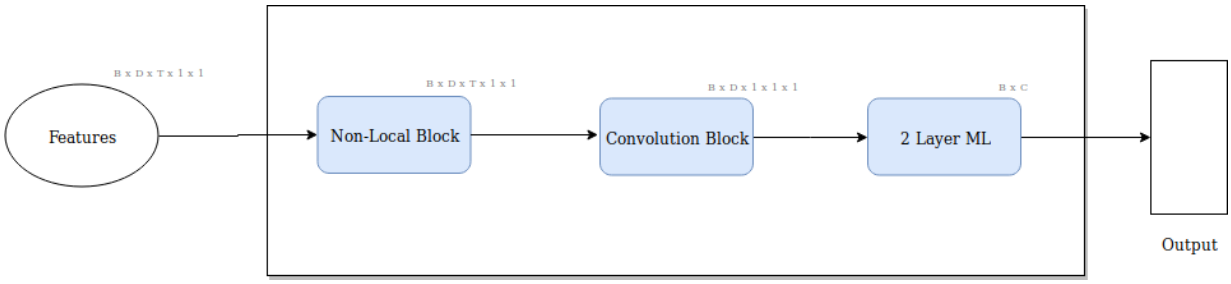

Figure 10: Non-Local Block **Temporal Alternative**

representations Z through two successive graph embedding layers. The outputted features are fed into a two-layer MLP with Dropout layers, Batch Normalization & Leaky-Relus nonlinearity. Figure 9 depicts the setup of network C.

4.3.4 Temporal feature alternatives

The networks described above consume the Spatio-temporal feature set where the features are of the form $f_i \in \mathbb{R}^{BxDxTxWxH}$ where W & H > 1. As described above, we want to investigate the networks ability to learn from only temporal features where the features will be of the form $f_i \in \mathbb{R}^{BxDxTxWxH}$ where W & H $= 1$. To achieve this, we have obtained the temporal feature set by average pooling over the W & H regions on the Spatio-temporal feature set. The temporal feature set can be fed to the same networks defined above with some minor tweaks. To adapt Network A and Network C for temporal features it is only a matter of parameter initialisation of the experiments. However, to adapt Network B it is required to remove the 3D Spatial pooling layer as this has already been done in the feature extraction phase of the temporal feature set. Figure 10 depicts the temporal alternative network without the 3D Spatial pooling layer.

5 Experiments & Results

In this section, we will describe how we set up our experiments as well as the results obtained. We will also perform an analysis of these experiments.

5.1 Experimental Setup

In the section above we have defined 3 networks with 2 variations of each, one consumes temporal features while the other consumes Spatio-temporal features. We train an additional baseline network on the temporal feature set. The network contains a 3D Average pooling layer over the time domain again with a kernel size of $(4, 1, 1)$ and a two-layer MLP classifier. This leads to 7 networks that we have to train & test. The networks described output logit probabilities of action classes for the features f_i representing a segment s_i with 32 frames. Therefore, since we are addressing the per-frame classification task, the prediction of the network for one segment s_i is associated with every frame in the segment.

We train these networks on the original train/test splits published with the Charades Dataset [SVW+16]. Training is performed by passing the logit probabilities through a Sigmoid activation which is used to calculate the Binary Cross Entropy Loss. The inputted features are fed in a Batch Size of 140 for 100 epochs over the dataset. After each epoch, the dataset is shuffled to avoid overfitting. The network is optimized using Adam optimizer with a learning rate of 0.0001. For testing, the logit probabilities are passed through a Softmax Activation which gives multi-class probabilities for the inputted segment. These probabilities are used the calculate the metrics described below on the test set after each epoch.

5.2 Metrics

The temporal localization problem, specifically on a dataset which contain multiple labels are usually evaluated on the Mean Average Precision (mAP) metric. This is a better metric to evaluate densely labeled videos in comparison to the Accuracy Metric. We investigate the correlation between the Average Precision (AP) per action class similar to what has been done in [SVW+16]. As we have discussed above, the Charades dataset is made up of vocabulary of 40 objects and 30 actions in 15 scenes. We will investigate any relation between these groups of action classes using both the mAP per group and the AP for each group member.

	SpaceTime mAP (%)	Time mAP (%)
I3D + 3D Pool + 2Layer MLP		5.13
I3D + Non-Local block + 2Layer MLP	11.58	9.408
I3D + ActionVLAD + 2Layer MLP	**14.31**	9.373
I3D + VideoGraphs + 2Layer MLP	13.76	**10.73**

Table 1: Results obtained from our methods with SpaceTime or Time featrues feed into our temporal model on the Charades dataset.

5.3 Temporal Localization

In this section we will investigate the results obtain on the temporal localization experiments described above using the Charades dataset.

5.3.1 Dataset Results

The networks defined in Section 4 were trained as described in Section 5.1, and are evaluated using the metrics described in Section 5.2. We train the networks for 100 epoch, save the model after each epoch and then evaluate the model on the test set. After training has finished, we pick the best performing model on the test set. Table 1 shows the results obtained on the best performing models. Our best performing model obtains 14.31% mAP when using SpaceTime features and 10.73% mAP on temporal features only. The best performing method on spatial-temporal features is ActionVLAD while VideoGraphs seem to to have a significant improvement over the other methods when dealing only with temporal features.

5.3.2 Temporal Models

The temporal models that we will be evaluating in this sections are those networks which were trained on the temporal feature set. We first look at the different performance between the three different temporal models. Using the mAP measure, we see that the VideoGraphs model performs better than the other models on average overall dataset.

Therefore, we have a look at the Average Precision of the different action classes. Figure 11 show the three temporal methods compared on a per-class basis. It is clear from the figure that VideoGraphs performs significantly better in most classes. Specifically, we observe better temporal performance in classes such as 'Sitting in a chair' & 'Working on a laptop'. Noticeably, this method does not perform well in action classes such as 'Closing a refrigerator' & 'Looking outside a window'.

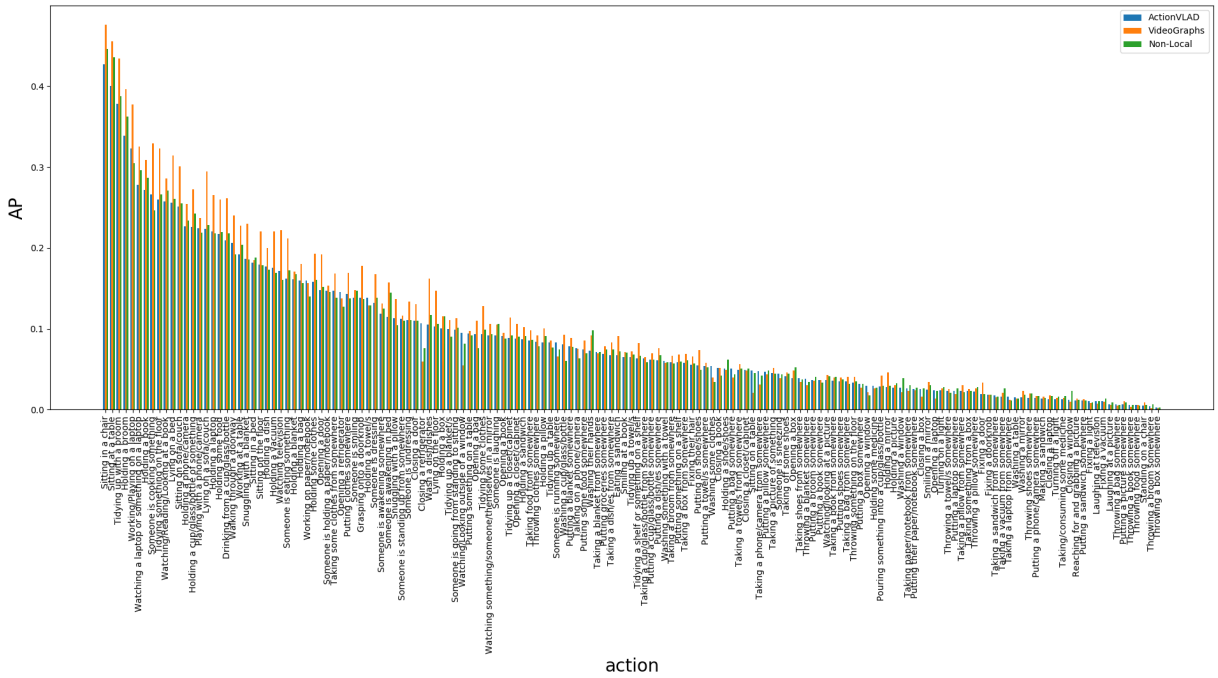

Figure 11: Temporal Model Average Precision (AP) per Action Class on the Charades dataset.

To further understand the generalization behaviour over the action classes we investigate the mAP over the 30 different verbs and 15 different objects contained in the vocabulary of the actions classes. We investigate the two best-performing methods VideoGraphs & ActionVLAD. In Figure 12 we plot the 5 highest and lowest AP difference for the two models. We can see that VideoGraphs perform reasonably better in 'Cooking', 'Lying' & 'Drinking' verb groups and 'Sofa/Couch', 'Floor' & 'Bed' object groups. While ActionVLAD performs reasonably better in the 'Run', 'Sneeze' & 'Close' verb groups and 'Refrigerator', 'windows and 'medicine' object groups.

5.3.3 Spatio-temporal models

Spatio-temporal models are those variations of the networks described above that are trained on the Spatio-temporal feature set. As we have seen above, the models perform differently when exposed to additional spatial information. Using the mAP measure over all the test set, we found that the ActionVLAD model performs best at 14.31%. Figure 13 depicts the Average Precision per-class for all action classes in the test set of the Charades dataset. ActionVLAD performs well on action classes such as 'Sitting in a chair' & 'Fixing

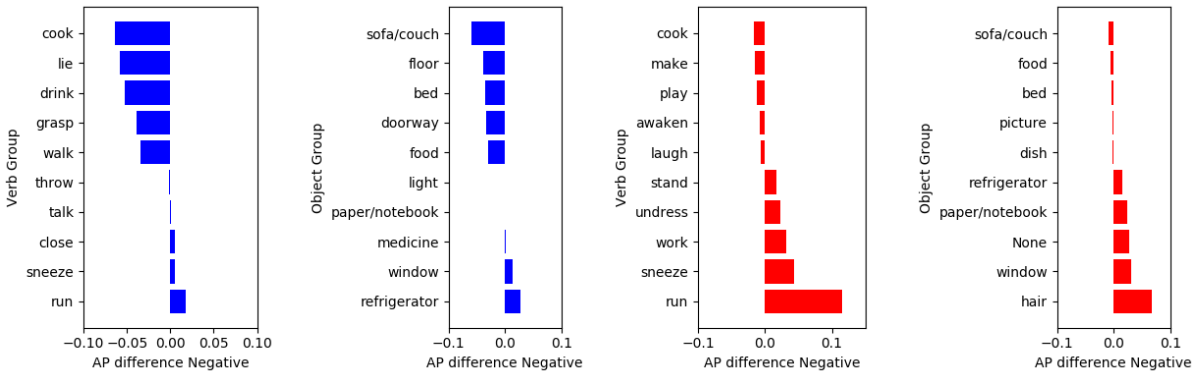

Figure 12: Comparison between Video Graphs (Negative) and ActionVLAD (Positive) on AP per class. Showing the top 5 positive and top 5 negative difference values. The Blue graphs indicate the models trained on the Temporal feature set while the Red graphs indicate the models trained on the Spatio-temporal feature set.

their hair'. An interesting observation is that VideoGraphs perform significantly better in actions such as 'Lying on the couch', 'Washing a mirror' & 'Washing a table'.

Figure 13, depicts the Average precision per class difference for the Spatio-temporal models on the ActionVLAD and VideoGraph models. We depict these models in red for verb and object groups. ActionVLAD performs significantly different than Node attention in the 'Run', 'Sneeze' & 'Work' verb classes while performing better in the 'Hair', 'Window' & 'None' object groups. Video Graphs performs slightly better in the 'Cook', 'Make' & 'Play' verb classes and the 'Sofa/Couch', 'Food' & 'Bed'.

5.4 Temporal vs Spatio-temporal feature sets

In this section, we will be investigating the behaviour of the two alternatives of each model ie. the model trained on the temporal feature set as opposed to training it on the Spatio-temporal feature set. We have shown above that there is a significant improvement when introducing the additional spatial information on the whole dataset. We will investigate the performance per class of the temporal vs Spatio-temporal models.

5.4.1 VideoGraphs

We plot the AP performance per-class for the top 50 performing classes for the VideoGraphs method in Figure 14. The scatter plot shows the 50 samples on a 2D plane where anything below the diagonal dotted line performs better when trained on the Spatio-temporal

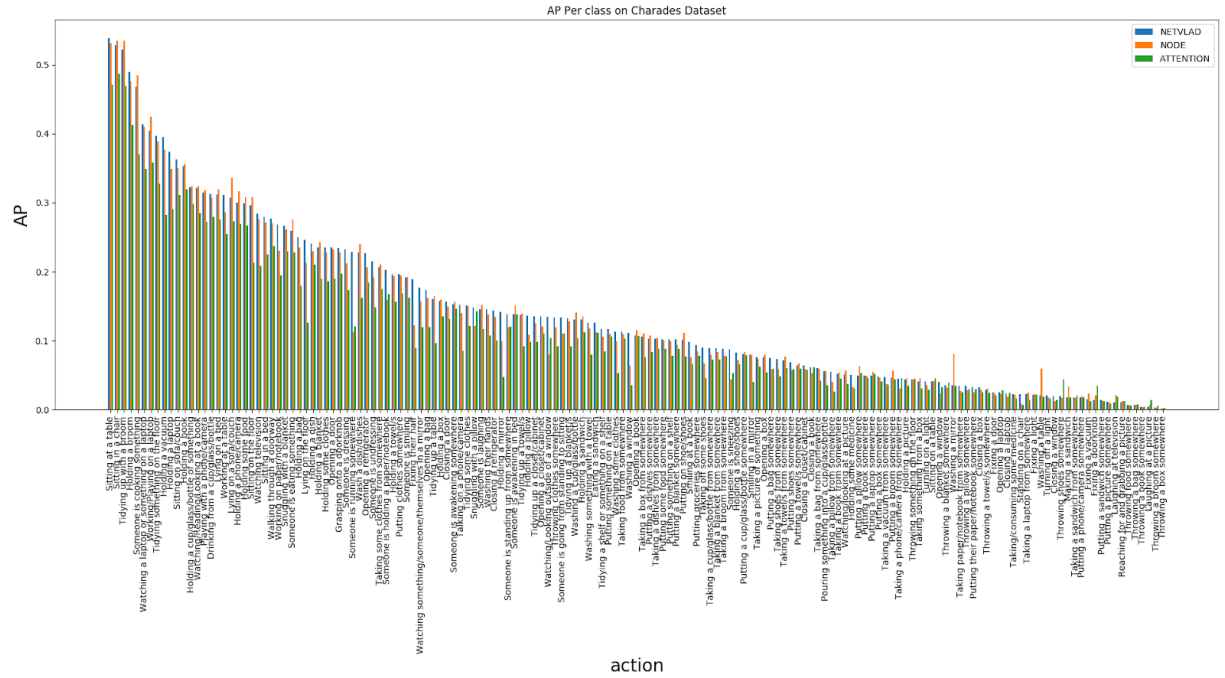

Figure 13: Spatio-temporal Model Average Precision (AP) per class for the Charades dataset.

features while if above, the model trained on temporal features performs better. When a sample is close to the diagonal line the method performs relatively the same on both methods ie. not gaining any extra information from the introduction of spatial features.

From this plot, we can see that 'Sitting in a chair', 'Sitting at a table' & 'Tidying with a broom' are the best performing classes similar to our previous analysis. We can also look at the classes which have high and low differences. 'Holding a vacuum', 'Someone is cooking' & 'Watching a laptop or something on a laptop' are the action classes which benefit the most from the addition of spatial information. While 'Lying in bed' & 'Someone is holding a paper/notebook' benefit very little from the addition of spatial information. In general, there is a very clear improvement in the top 50 performing classes when trained on Spatio-temporal features.

Figure 14 shows the per class

The scatter plot above gives us some insight into the improvement of VideoGraphs when trained on the Spatio-temporal feature set. However, we plot another scatter plot of the AP performance of the verb and object groups. These are plotted on the same plane and can be found in Figure 15. The green points depict Verb groups and the red points depict the object groups. We observe that there are some high performing groups which

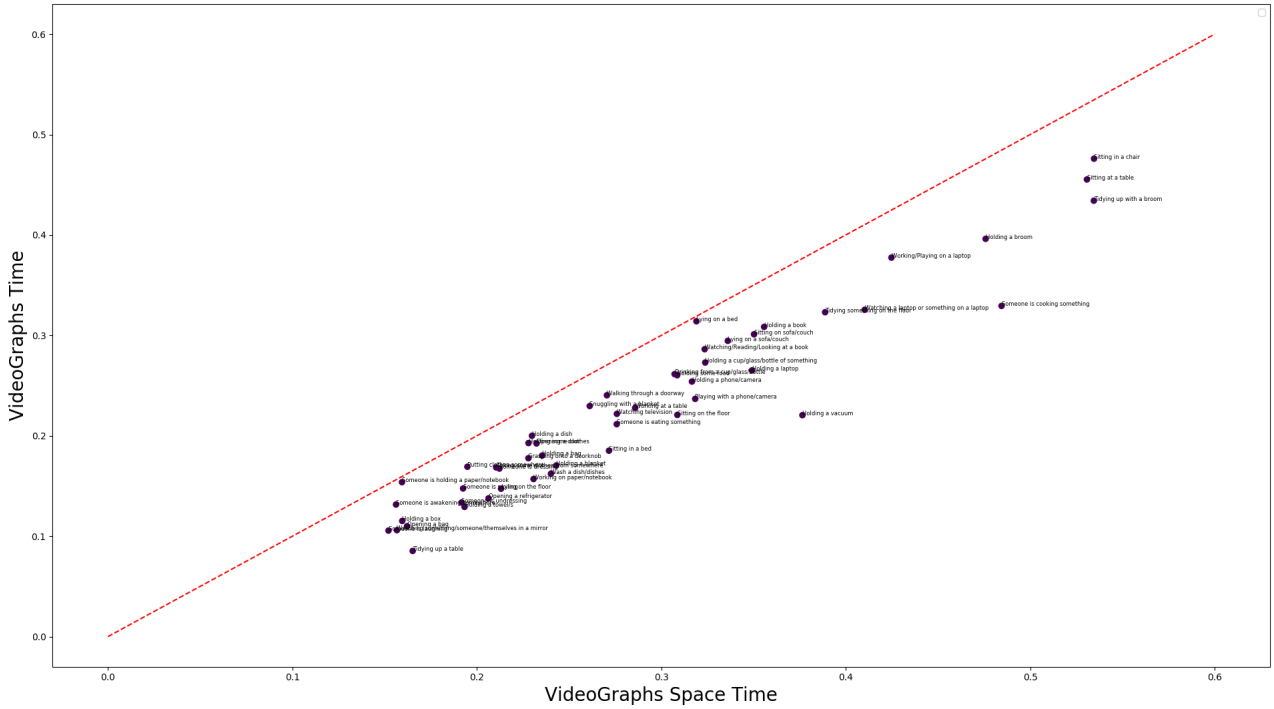

Figure 14: Comparing the **Top 50** performing action classes of the VideoGraphs networks.

are significantly improved by the introduction of spatial features mainly the 'Cooking' verb class & the 'hair' & refrigerator' object classes. The 'Stand', 'Awaken' & 'Snuggle' classes are almost not impacted at all by the introduction of these features.

5.4.2 ActionVLAD

Subsequently, we plot the per-class performance for the top 50 performing class for the ActionVLAD method. As before the scatter plot compares the Temporal and Spatio-temporal trained models. We observe that the action classes which perform significantly better with the addition of spatial information are the same as the VideoGraphs method. There are not many action classes that have similar performance to the Temporal alternative in the top 50 classes. However, we can see that in general, the best performing classes are shifted more toward the Spatio-temporal model indicating that in general, this method generalises better on these classes.

As we have done for VideoGraphs, we plot a scatter plot of the verb groups and object groups on the same plane for the ActionVLAD model seen in Figure 17. Again we observe the 'cooking' verb groups performing significantly well with the introduction of spacial information as well as the 'running' group. 'Sofa/Couch', 'Floor' & 'Vacuum' are some of

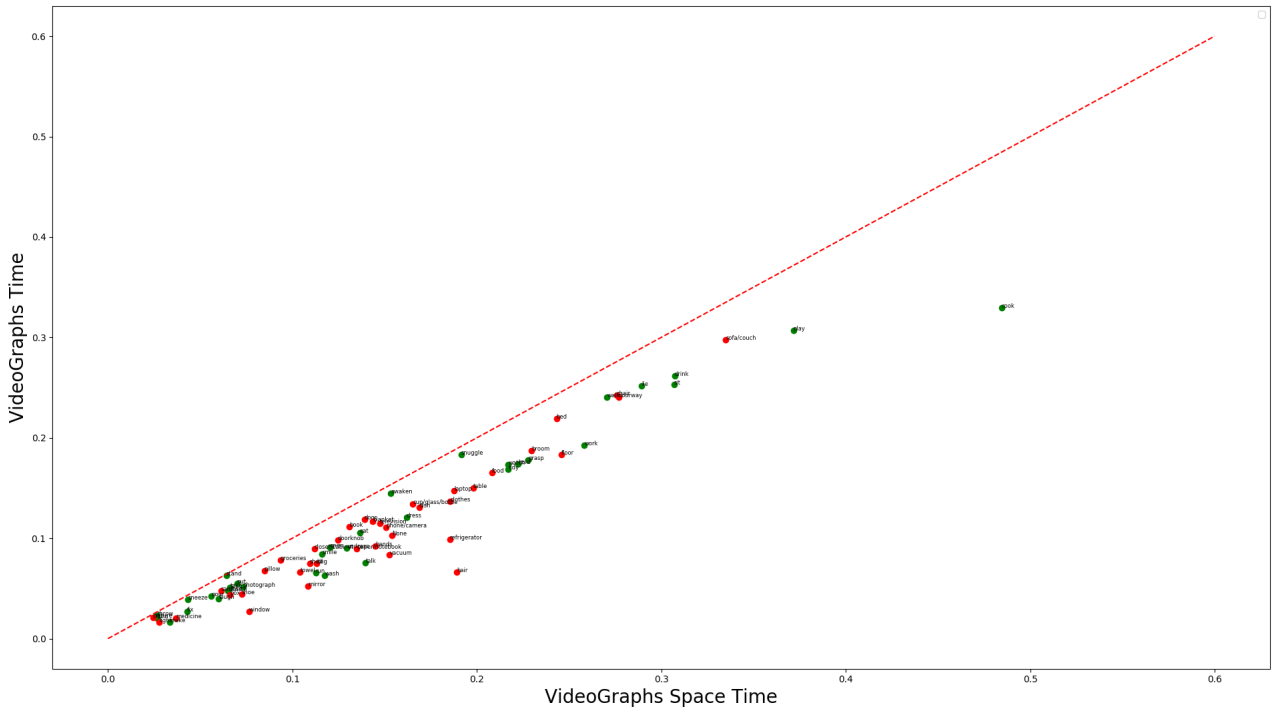

Figure 15: Scatter-plot of the Average Precision per Verb group (**Green**) & Object group (**Red**) for the VideoGraphs temporal vs Spatio-temporal models

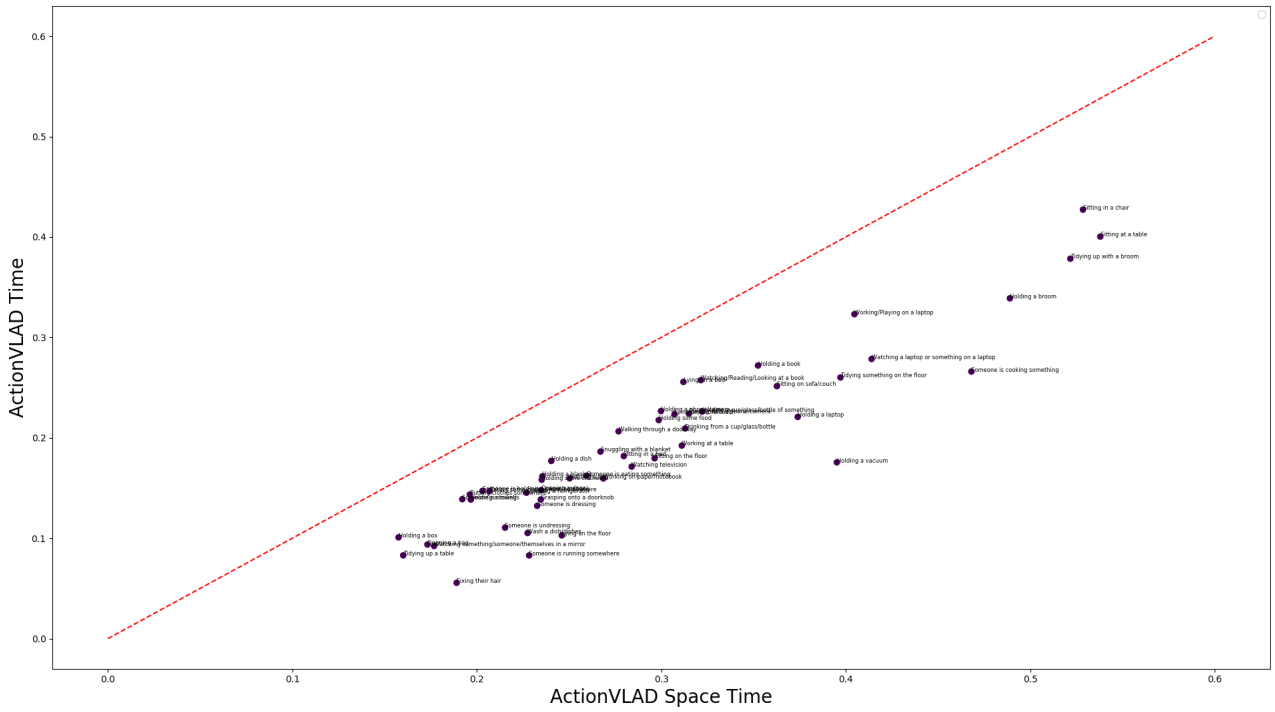

Figure 16: Comparing the **Top 50** performing action classes of the ActionVLAD networks.

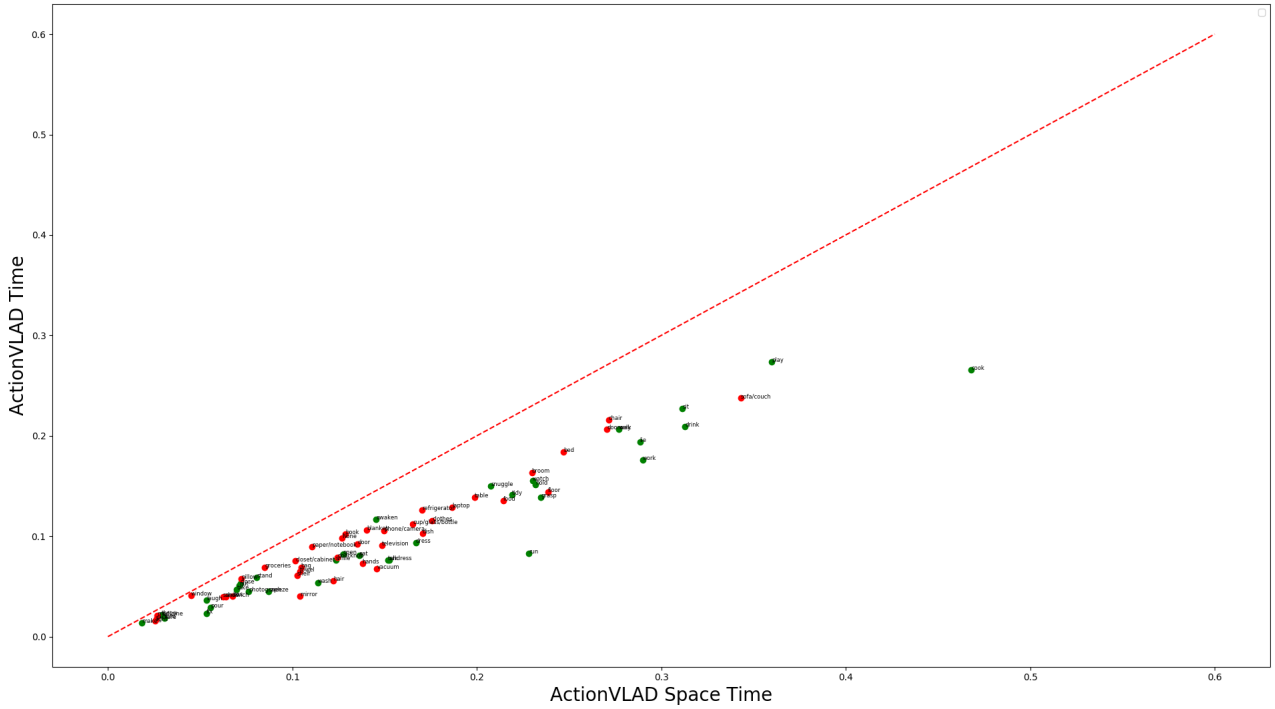

Figure 17: Scatter-plot of the Average Precision per Verb group (**Green**) & Object group (**Red**) for the ActionVLAD temporal vs Spatio-temporal models

the object group classes that perform relatively well. Not many groups for this method perform similar to each other in the higher value region.

5.4.3 Non-Local Block

Finally, we investigate the relationship of the Non-Local block model trained on temporal features and Spatio-temporal features. We do this by a scatter plot as we have done above, depicted in Figure 18. An interesting insight from this figure is that the action class 'Someone is awakening' performs slightly better with the model trained on temporal features. Nonetheless, the majority of the action classes benefit from the addition of spatial information. Specifically, Holding a vacuum', 'Someone is cooking something' and 'tidying up with a broom' action classes benefit significantly from the spatial information.

Furthermore, we group the AP per verb and noun group to observe the behaviour in a more general level depicted in Figure 19. From the AP per class, we had observed that the action class 'Someone is awakening' performs better on the model trained on temporal features only although the verb group 'awaken' which it forms part of performs slightly better with spatial information. We observe again that the 'Cooking' verb group performs

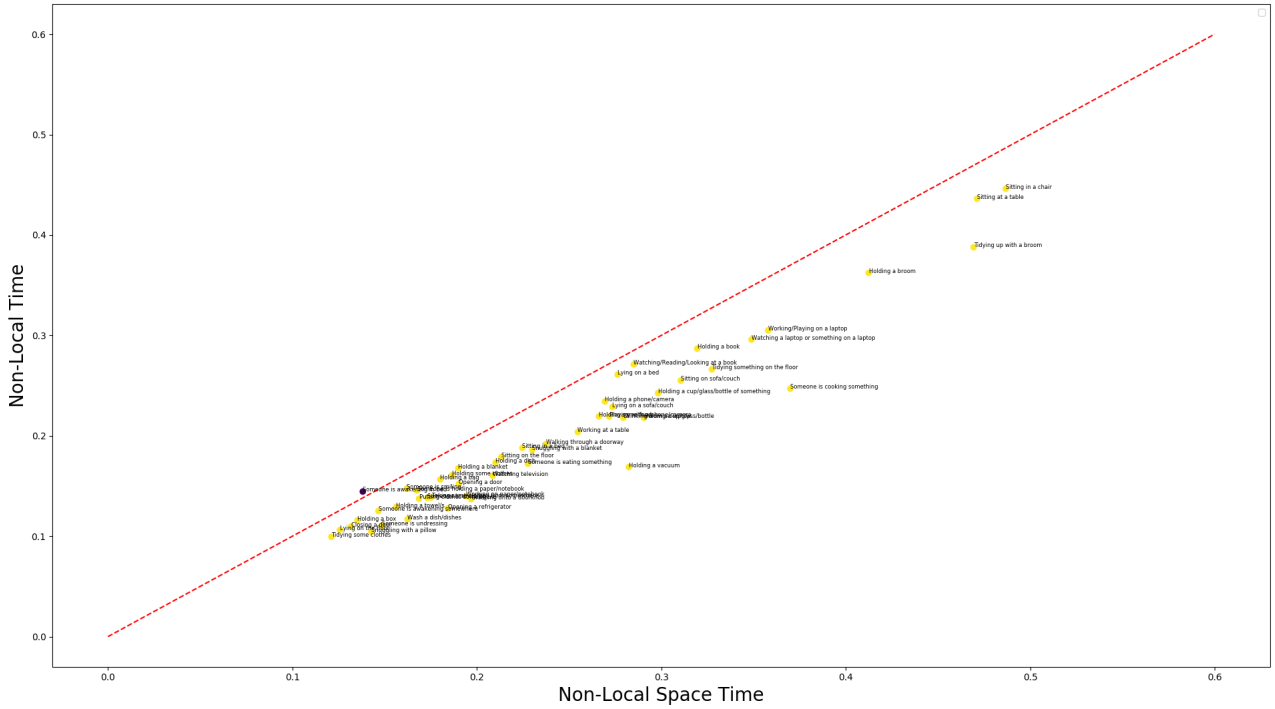

Figure 18: Comparing the **Top 50** performing action classes of the Non-Local block networks.

significantly better in the Spatio-temporal model.

5.5 Visualization of image regions with high activations

From the analysis above, we can observe that these methods outperform each other on specific action classes and/or sub-groups of action classes. Therefore, we take a closer look specifically at the ActionVLAD & VideoGraphs Spatio-temporal models. As we have discussed, the ActionVLAD layer performs soft-assignment over cluster centroids while the VideoGraphs method learns node representations. To some extent, we can think of the ActionVLAD centroids and the VideoGraph nodes as learnt latent representations of visual elements. To fully understand these learnt representations, we look at a way to visualise them.

To visualise these nodes/centroids, we look at a way to backtrack high activations to an image and it's respective regions. We achieve this by using the Spatio-temporal features we have extracted in our experiment phase which are of the form $f_i \in \mathbb{R}^{BxCxTxWxH}$. With the hyper-parameters of our experiments specifically, our feature vectors are of the form

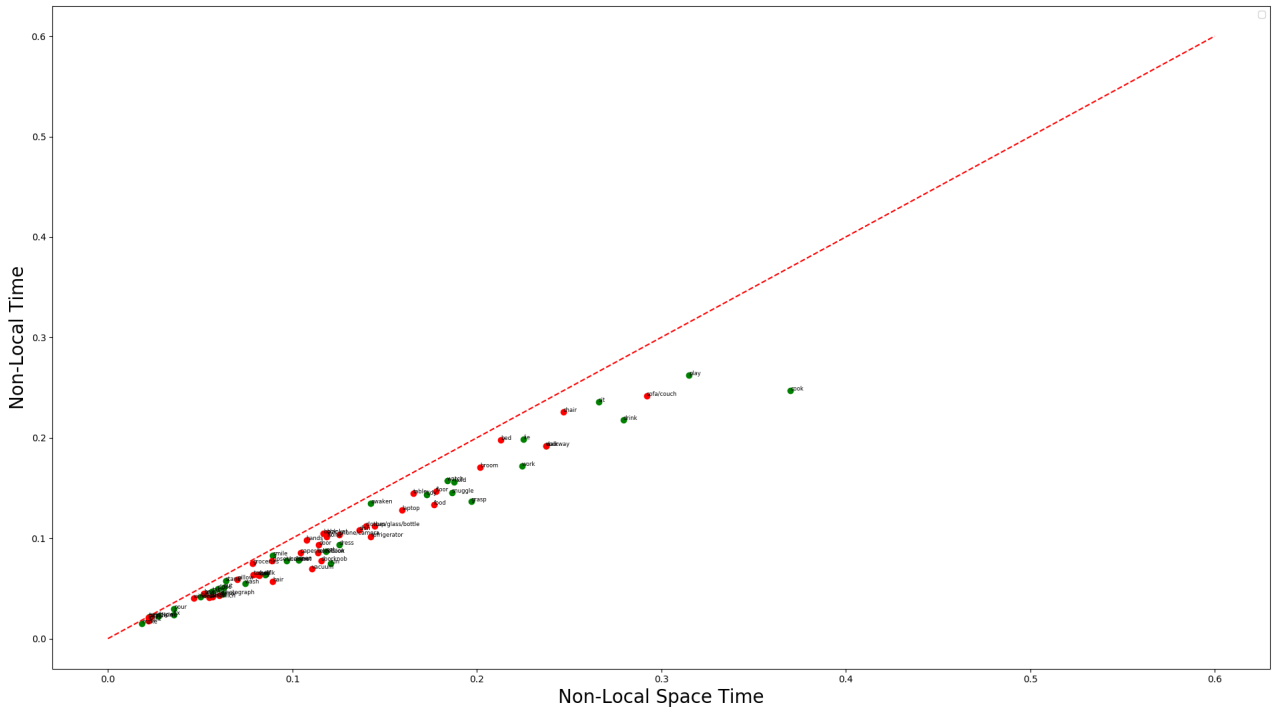

Figure 19: Scatter-plot of the Average Precision per Verb group (**Green**) & Object group (**Red**) for the Non-Local Block temporal vs Spatio-temporal models

$f_i \in \mathbb{R}^{Bx1024x4x7x7}$ where 4 is the timestep, which represents an aggregation in groups of 8 frames performed by the I3D network

where T is the temporal dimension set at 4 containing an aggregation of features from the 32 input frames grouped at a rate of 8 frames. The W & H dimension set both at 7 in our experiments, represent the 8 successive images in a 7x7 grid ie. 49 regions. When these features are fed into the network, we save the soft-assignment activations of ActionVLAD and the similarity values from the Node Attention layer of VideoGraphs. This will allow us to pinpoint, regions within these groups which have the highest similarity to nodes/centroids. We keep only activations which have a 95% similarity. To finally visualise these regions, we take the middle frame from the group with the highest activation & draw a box around the image regions with 95% similarity or more. We perform this for all images in the dataset and investigate some of these visualizations. Some clusters had a very high number of samples obtained while others did not produce any outputs even at a threshold of 75%. We have identified patterns in some of these visualizations.

The first node we visualise is node 25 from the ActionVLAD model trained on Spatio-temporal features. This node had the highest amount of samples generated with a threshold

of 95% similarity. From a manual investigation of the generated regions, we recognise that most of the images contain individuals utilising a cup or something to hold water. With these observations in mind, we intuitively associate this with the 'cup/glass/bottle' object class. This object class is used in conjunction with the 'Drink', 'Hold', 'Pour', 'Put', 'Take' & 'Wash. From the samples, we can see that this centroid contains high activations when exposed to regions of 'Drinking from a cup/glass/bottle'. Some samples of the images and their respective high activation region are displayed in the top row of Figure 20

Our second node to visualise from the ActionVLAD model trained on Spatio-temporal features is node 28. By manual investigation of the generated images/regions, we identify that the regions with close similarity to this node contain individual utilising a laptop device. Therefore, the object class that seems to be similar to this node is probably the 'laptop' object group. The 'laptop' object group is used in conjunction with the follow verb groups 'Close', 'Hold', 'Open', 'Put', 'Take', 'Watch' & 'Work'. From observing the generated images, we can see numerous clear samples of 'Holding a Laptop', 'Working/Playing on a Laptop' & 'Closing a laptop' amongst others. Some of these samples are depicted in the middle row of Figure 20.

The final node we visualise from the ActionVLAD model trained on Spatio-temporal features is node 20. Again we manually investigate the generated images and identify and that the samples contain mostly interactions with a handheld device mainly, controllers, phones and cameras. The object class that is related to this is the 'phone/camera', interestingly enough there are no object class related to a game console controller, yet, it is still closely associated with this node. The 'phone/camera' object class is closely associated with verb groups such as 'Hold', 'Play', 'Put', 'Take' & 'Talk'. We can see clear examples of 'Holding a phone/camera', 'Playing with a phone/camera' & 'Talking on a phone/camera'. Similarly, we investigate the image regions which have a high activation with the nodes from the VideoGraphs model trained on the Spatio-temporal feature set. The first Node we investigate is node 37 which containing many samples of individuals cleaning an object. We assosicate this node with the 'wash' verb groups which is used in the vocabulary of the action classes with the clothess', 'towel', 'window', 'mirror', 'cup/glass/bottle' & 'hands' object classes. We observe instances which we can identify was 'Washing something with a towel', 'Washing a cup/glass/bottle' & 'Washing some clothes'. These samples can be observed in the top row of Figure 21.

Finally, we also investigate Node 1 from the VideoGraphs models trained on the Spatio-temporal feature set. The samples obtained contained actors lying or sitting on several

Figure 20: **Action VLAD** Node Representation. **Top** row contains representation of Node 25. **Middle** row contains the representation of Node 28 while the **Bottom** row contains representations of Node 20.

objects in the scenes. This node seems to be representing more than one verb group and object group. The 'lying' & 'sitting' verb classes are used in conjunction with similar object classes such as 'chair', 'floor', 'bed' and 'sofa/couch'. We observed numerous action classes in the high activation regions of samples generated such as 'Sitting in a chair', 'Lying on a bed' & 'sitting on the sofa/couch'. Some of samples are depicted in the bottom row of Figure 21.

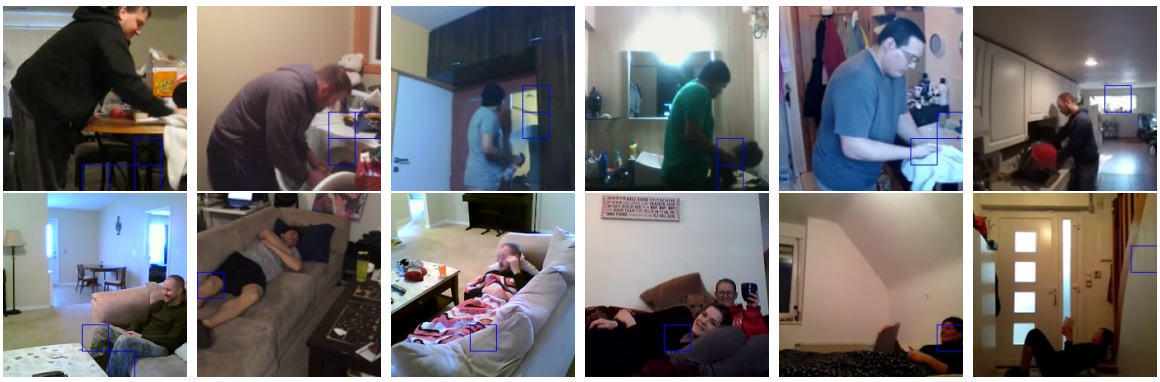

Figure 21: **Video Graphs** Node Representation. **Top** row contains representation of Node 84 while the **Bottom** row contains representations of Node 1.

6 Discussion

In this section we attempt to answer the research question presented in the beginning of the research. '*Are ActionVLAD, Self-Attention & VideoGraph methods amenable to temporal action localization in untrimmed videos?*', '*Specifically, can we use these methods as part of a model to temporally localize actions in videos on a per-frame level rather than video-level?*' and '*Given that we extract a temporal feature set and a Spatio-temporal feature set from a video. Do models trained on the temporal feature set generalize differently than the models trained on the Spatio-temporal feature set?*'.

The first question we attempt to answer is if ActionVLAD, Self-Attention & VideoGraph methods are amenable to temporal action localization in untrimmed videos. This question can easily be answered by the pipeline we have set up to incorporate these methods into layers for a temporal localization network. We have successfully trained & evaluated these networks with common practise techniques from research. We have successfully shown that these methods perform well when applied to action localization of untrimmed videos.

Our second question related to specifically including these methods in models to localize actions on a per-frame level rather than video-level. Again the pipeline we developed included these methods in a temporal localization network and was trained and evaluated successfully. Secondly, we achieve per-frame classification by classifying every 32-frame segment over the entire dataset. Furthermore, these segment-level classifications are assigned to the 32 individual frames.

To continue to understand if these methods learn any semantic representations from our dataset, we visualised highly similar regions of images to centroids/nodes of the ActionVLAD & VideoGraphs methods. We have seen various examples of visualizations that represent a more general representation than the action labels themselves. We have shown through training & evaluating these models on a frame-level temporal localization dataset that these models work well on the temporal localization problem.

We attempt to answer the final question relating to the different performance of the model trained on either a temporal or Spatio-temporal feature set. We have shown in our model the feature extraction step which includes the steps taken to extract such feature sets from a video dataset. We show that some methods perform better than others when exposed to different feature sets. Specifically, we found that while ActionVLAD was the best performing method when trained on the Spatio-temporal feature set, VideoGraphs performed significantly better than ActionVLAD on the temporal feature set.

Additionally, we show that on the 157 labelled classes some methods perform slightly better with temporal features while the majority performs better with the Spatio-temporal features. On the other hand, when looking at the performance on a verb and object level we found that some networks perform better in some groups than others and that all groups perform equally or better than when trained on temporal features. However, there is some classes highlighted above, that benefit little from the addition of the spatial information. For example, we have seen that the VideoGraphs method perform well in 'cook' group in both temporal and Spatio-temporal models while the 'run' group performs significantly better on both ActionVLAD networks. This can be seen in the scatter plots above and the horizontal bar graphs (Figures 12 & 14-19).

7 Conclusions

We successfully answered the three research questions we have presented in this research by showing that ActionVLAD, Self-Attention & VideoGraphs are methods which can be used for temporal localization of action in untrimmed videos. We have achieved this by developing several temporal localization networks that use pre-trained 3D CNNs to extract features from untrimmed videos and classify these features into actions. As well as being able to perform a per-frame level classification of such video files. We designed 3 networks that incorporate these methods as layers with two alternatives of each, one consuming temporal only features and the other consuming Spatio-temporal features. We found that ActionVLAD performs significantly better on the Charades dataset with 14.31% mAP when trained on the Spatio-temporal feature set. While VideoGraphs obtains a 10.73% mAP which performs better than the other methods when trained on the temporal feature set. We compared these methods on a different grouping of labels and found some methods perform better in certain groups. To further analyse the behaviour of the models, we visualized image regions close to latent representations these methods learnt.

References

[AGT+16] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[BD01] Aaron F Bobick and James W Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (3):257–267, 2001.

[BMW+11] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. Sequential deep learning for human action recognition. In *International workshop on human behavior understanding*, pages 29–39. Springer, 2011.

[CZ17] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. *CoRR*, abs/1705.07750, 2017.

[DAHG+15] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[DPVG16] Ali Diba, Ali Mohammad Pazandeh, and Luc Van Gool. Efficient two-stream motion and appearance 3d cnns for video classification. *arXiv preprint arXiv:1608.08851*, 2016.

[DRCB05] Piotr Dollár, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. Behavior recognition via sparse spatio-temporal features. VS-PETS Beijing, China, 2005.

[GHS11] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Actom sequence models for efficient action detection. In *CVPR 2011*, pages 3201–3208. IEEE, 2011.

[GHS13] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Temporal localization of actions with actoms. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2782–2795, 2013.

[GRG+17] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 971–980, 2017.

[GSV+17] Chunhui Gu, Chen Sun, Sudheendra Vijayanarasimhan, Caroline Pantofaru, David A. Ross, George Toderici, Yeqing Li, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. AVA: A video dataset of spatio-temporally localized atomic visual actions. *CoRR*, abs/1705.08421, 2017.

[HGS19] Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Videograph: Recognizing minutes-long human activities in videos. *arXiv preprint arXiv:1905.05143*, 2019.

[HHP17] Samitha Herath, Mehrtash Harandi, and Fatih Porikli. Going deeper into action recognition: A survey. *Image and vision computing*, 60:4–21, 2017.

[Hog83] David Hogg. Model-based vision: a program to see a walking person. *Image and Vision computing*, 1(1):5–20, 1983.

[HS+88] Christopher G Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

[JDSP10] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *CVPR 2010-23rd IEEE Conference on Computer Vision & Pattern Recognition*, pages 3304–3311. IEEE Computer Society, 2010.

[JXYY12] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012.

[KCS+17] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[KTS+14] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[Lap05] Ivan Laptev. On space-time interest points. *International journal of computer vision*, 64(2-3):107–123, 2005.

[LQY+16] Qing Li, Zhaofan Qiu, Ting Yao, Tao Mei, Yong Rui, and Jiebo Luo. Action recognition by learning deep multi-granular spatio-temporal video representation. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, ICMR '16, pages 159–166, New York, NY, USA, 2016. ACM.

[MHS09] Pyry Matikainen, Martial Hebert, and Rahul Sukthankar. Trajectons: Action recognition through the motion analysis of tracked features. In *2009 IEEE 12th international conference on computer vision workshops, ICCV workshops*, pages 514–521. IEEE, 2009.

[MPK09] Ross Messing, Chris Pal, and Henry Kautz. Activity recognition using the velocity histories of tracked keypoints. In *2009 IEEE 12th international conference on computer vision*, pages 104–111. IEEE, 2009.

[MSPGiN16] Alberto Montes, Amaia Salvador, Santiago Pascual, and Xavier Giro-i Nieto. Temporal activity detection in untrimmed videos with recurrent neural networks. *arXiv preprint arXiv:1608.08128*, 2016.

[NDL+05] Feng Ning, Damien Delhomme, Yann LeCun, Fabio Piano, Léon Bottou, and Paolo Emilio Barbano. Toward automatic phenotyping of developing embryos from videos. *IEEE Transactions on Image Processing*, 14:1360–1371, 2005.

[OMK+14] Sangmin Oh, Scott McCloskey, Ilseo Kim, Arash Vahdat, Kevin J. Cannons, Hossein Hajimirsadeghi, Greg Mori, A. G. Amitha Perera, Megha Pandey,

and Jason J. Corso. Multimedia event detection with multimodal feature fusion and temporal concept localization. *Machine Vision and Applications*, 25(1):49–69, Jan 2014.

[PR18] AJ Piergiovanni and Michael S Ryoo. Learning latent super-events to detect multiple activities in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5304–5313, 2018.

[SCZ+17] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5734–5743, 2017.

[SJYS15] Lin Sun, Kui Jia, Dit-Yan Yeung, and Bertram E Shi. Human action recognition using factorized spatio-temporal convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4597–4605, 2015.

[SVW+16] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. *CoRR*, abs/1604.01753, 2016.

[TBF+14] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. C3D: generic features for video analysis. *CoRR*, abs/1412.0767, 2014.

[TCLZ12] Y. Tian, L. Cao, Z. Liu, and Z. Zhang. Hierarchical filtered motion for action recognition in crowded videos. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(3):313–323, May 2012.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[WFF+18] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krähenbühl, and Ross Girshick. Long-term feature banks for detailed video understanding. *arXiv preprint arXiv:1812.05038*, 2018.

[WGGH18] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[WRB11] Daniel Weinland, Remi Ronfard, and Edmond Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer vision and image understanding*, 115(2):224–241, 2011.

[WTVG08] Geert Willems, Tinne Tuytelaars, and Luc Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *European conference on computer vision*, pages 650–663. Springer, 2008.

[YHNHV+15] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702, 2015.

[YRJ+18] Serena Yeung, Olga Russakovsky, Ning Jin, Mykhaylo Andriluka, Greg Mori, and Li Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *International Journal of Computer Vision*, 126(2-4):375–389, 2018.

[YS05] Alper Yilmaz and Mubarak Shah. Actions as objects: A novel action representation. CVPR, 2005.