UNIVERSITY OF OTTAWA

DOCTORAL THESIS

Efficient, Movement-Based Skeleton Action Recognition

Author:
Nicolas FLEECE

Supervisor:
Dr. Robert LAGANIÈRE

A thesis submitted in fulfillment of the requirements
for the degree of Master of Computer Science, Specialization in Applied AI

in the

VIVA Research Lab
School of Electrical Engineering and Computer Science

July 19, 2023

UNIVERSITY OF OTTAWA

Abstract

Faculty of Engineering

School of Electrical Engineering and Computer Science

Master of Computer Science, Specialization in Applied AI

Efficient, Movement-Based Skeleton Action Recognition

by Nicolas FLEECE

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too...

"Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism."

Dave Barry

Acknowledgements

The acknowledgments and the people to thank go here, don't forget to include your project advisor…

Contents

List of Figures

List of Tables

List of Abbreviations

LAH	**L**ist **A**bbreviations **H**ere
WSF	**W**hat (it) **S**tands **F**or

Chapter 1

Introduction

People interact with their environment in unique and nuanced ways, and throughout our lives, humans have learned to identify and categorize the different actions that we perform.

1.1 Action Recognition

For humans, the problem of Human Action Recognition is rather simple. We use past experiences throughout childhood and adult life to be able to pick out the various ways a person moves, and translate that into a familiar action that we have seen before. Combine that with objects that a person may be interacting with, and humans are remarkably good at discerning what actions other humans are involved in. However, as is with most things in the domain of computer vision, this ability does not translate well into the realm of artificial intelligence. The slightly different ways that people may perform these tasks add a layer of complexity that is difficult for a model to overcome.

1.2 Applications

Security is perhaps the most obvious example of action recognition usage. Security personnel are constantly on the lookout for suspicious individuals that may be of concern, or who are performing illegal actions. This can be as simple as trying to find those who are shoplifting in stores, where the CCTV footage can be used live to find those who are actively stealing from stores. It could also be something more complex, such as security checkpoints in airports, where screening officers are

constantly watching for suspicious individuals. In this case, a system capable of analyzing the way every person acts and pointing out those who it sees as suspicious could greatly assist those and point them in the correct direction.

Health Care is a slightly different, but nonetheless very interesting application of action recognition. A very large part of how action recognition can help those in the healthcare field is use in monitoring those who need round the clock care, primarily the elderly. If an elderly person chooses to live at home, the action recognition model may allow healthcare workers to, at a distance, manage many people and focus their attention on those who have been flagged as in danger of injury. This can often be done by very lightweight models [1].

Video Summarization is perhaps not directly related to action recognition, but rather action recognition is a very useful part of video summarization. If you must summarize a video where the main subjects tend to be humans, a large part of figuring out what is going on in the video is figuring out what action the person is performing, for example, for a summary to be something such as '*The person is fishing.*', the model must have some understanding of what fishing is. Similarly, if the main subject of the video is not a person, it may still be useful to know what those in the background are doing, for example '*A dog is sitting on a bench, there are people doing yoga in the background*', the model again must have an idea of what yoga is, and how a person performs said actions.

1.2.1 Ethical Issues

As with most applications of artificial intelligence, computer vision cannot be researched and discussed without taking into account the ethical issues that surround it. With AI being such a quickly evolving space, it is crucial that any researchers be aware of these issues. In this section, I will focus particularly on how it may affect action recognition, touching on other areas of AI in general to further illustrate my points.

Privacy can become a concern in many areas, the healthcare example given previously in this chapter is one of the most obvious. With elderly people, often one of the draws to staying in their own private homes is the privacy that it offers, if the action recognition model is to be used to ensure that they remain safe, it must

come with some removal of privacy. There is also a question of what happens to the data of a person who is using this kind of service, since it is almost certainly sent to a server to be processed given the typical size of these models, what kind of data retention policies might this company have in place, are they sharing this data with others, or is the data going to be used to further train. These are all issues that often follow AI since the training of models requires such a massive amount of data, and in action recognition this can contain people who are not necessarily aware of their data being used in such ways.

Bias is one of the most common ethical issues in AI that can appear. In artificial intelligence, and computing in general, the principle of "garbage in, garbage out", is a common one to illustrate that if the inputs into a model are not of high quality, the outputs will not be of high quality. This can often be the case in datasets where say a group of people are not accurately represented, and while the model itself is not discriminatory, it will follow the data it has been given. Take the example previously discussed of airport security. Airport security has been scrutinized in the past for singling out individuals of particular races or who look a particular way. This may mean that if a model is being constructed that searches for people who may be flagged later in security, the majority of positive flags that were screened further would be of this group of people. The resulting dataset that is constructed would be biased against this group of people, therefore resulting in a model similarly biased. This type of issue has been shown in many different areas, another example being speech recognition models used by voice assistants not being able to recognize particular accents as they were not represented in the original dataset. These kinds of reasons are why it is crucial for researchers to be aware of and study their datasets when it comes to human data to avoid these biases and ensure that their data is well balanced.

Transparency is a rather difficult, and often nearly impossible problem to solve with modern AI models. Given the fact that these models at minimum have millions of parameters that all contribute to the complex calculations towards the output, deciphering exactly how they work and make their decisions is difficult. These models are often depicted as black boxes, where the only context we are allowed is what inputs and outputs, and nothing in between. In the cases of something such as an

airport security checkpoint, the model may mark a person as acting suspicious in a line of passengers. The model is not able to specifically express what made the passenger appear suspicious, and it may even be incorrect in it's assumptions. This means that the officer who is reviewing the flags set by the model has to make a decision that leads to one of two possible scenarios:

1. The model is correct, but cannot communicate it's exact reasoning with the officer, the officer does not see what the model sees and a potential threat is ignored

2. The model is incorrect, but the officer thinks that he sees something, and a person who is not a threat is put through unnecessary screening, and other potential threats may not be screened

1.3 Challenges

Human action recognition is a very difficult task that comes with many issues, some of which continue to be major challenges moving forward with very complex modern models.

Backgrounds often not static in videos. Often they contain a lot of data that is ever changing and often can contain other secondary subjects performing actions that may not be relevant to the subject that we are trying to determine the action of. While humans are very good at focusing on the person who is performing the action and ignoring things occurring in the background enough to not get confused. AI models do not have this inherent ability and often can get confused from background changes, and effectively must both identify the person and determine what action they are performing within the same model. This can be further worsened by any camera movement, resulting in both the subject moving throughout the frame, but the background can completely change with a 90 degree camera angle change

1.4 Problem Definition

The problem of human action recognition is defined as taking a video of a person performing a particular action, and passing it through a model to determine the

specified action the person is performing. Pose-based action recognition is the problem of performing this detection primarily using the skeleton data of the people in the frame.

1.5 Thesis Structure

This thesis will begin by exploring the structures of typical convolutional neural networks in chapter 2, and more research relevant to action recognition in chapter 3. A new novel representation is proposed in chapter 4, and the experiments and their results are detailed in chapter 5. Finally the conclusions and final takeaways of the thesis are presented in chapter 6.

Chapter 2

Convolutional Neural Networks

2.1 Structure

2.2 Kernel

2.2.1 3 Dimensional Convolutions

2.3 Classic Architectures

2.3.1 AlexNet

2.3.2 VGG-16

2.4 Modern Architectures

2.4.1 ResNet

2.4.2 Residual Attention Network

2.5 LSTMs

Chapter 3

Literature Review

3.1 Image Classification

Image classification is the precursor problem to action recognition. Without the ability to classify individual images, the ability to classify videos, which functionally are a list of individual images, would never have been researched as nearly every technique for action recognition can be tracked from some form of image classification task.

An example of this task is the CIFAR dataset [2], where the goal is to simply classify relatively low resolution images into either 10 or 100 classes, depending on the version of the dataset. This dataset is very well explored, and results have high 90% accuracy values through very complex models and modern techniques. However, some simple models such as EfficientNet [3] are able to achieve this above 90% metric while providing a model that is efficient and able to train and evaluate images quickly. The popularization of image classification led to the popularization of CNN's which are also actively used in action recognition, and particularly popular in intermediate representation models that will be discussed in section 3.7.1.

3.2 Optical Flow

Optical Flow, not dissimilar to pose estimation described in section 3.6, is something that is often taken for granted in models that utilize it, however it can be computed in several different ways. An example of an optical flow technique is the method utilized by the I3D model [4] which will be discussed later in this chapter. This technique is known as TV-L optical flow [5], and utilizes a formula based on the

total variation (TV) regularization and the robust L^1 norm in the data fidelity term. While the methods described in this paper and many others are rather complex, the idea that provides performance improvements is that optical flow represents the movement of a person from one frame to another and eliminating background information which does not move, this is further demonstrated in figure 3.1.

(a) First frame (b) Second frame (c) Optical flow field

FIGURE 3.1: An example of the optical flow field **(c)**, resultant from the first and second frames **(a) (b)**. The goal being that the background is not in the optical flow field, only the movement of subjects in the frames are considered. This particular example is utilizing the TV-L optical flow [5].

3.3 CNN Based Models

Naturally, the first approach to feeding video data into models is to process the raw RGB frames. This technique is derived from image classification tasks, where lightweight and relatively simple CNN's have been shown to have good performance when classifying single images. The logic would then follow that these types of models may be able to classify videos, however they must be modified a few different ways, which will be described further in this chapter.

3.3.1 CNN + LSTM Models

In very classic models, utilizing existing CNN architectures is a very simple process. The individual frames are passed through the CNN model, producing a feature map for each frame. These feature maps are then pooled and passed into dense layers which produces an output. While very simple and fast, this model completely ignores any temporal activity, meaning that the model cannot determine how a person moves throughout a video from one frame to another. This would make differentiating some reversible actions such as running forwards vs running backwards very difficult.

Figure 3.2 shows the typical modern structure for this solution. After the features are extracted from each of the 2 dimensional CNN, they are passed through a LSTM. The goal of this LSTM module is to carry features from one frame to another and add some temporal element.

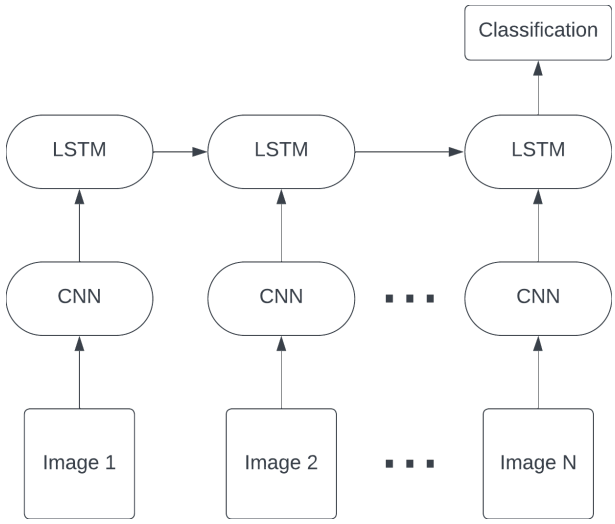

FIGURE 3.2: An example structure of a simple CNN-LSTM based model, each individual frame being individually fed into the CNN, and then passed to a LSTM.

The advantages of this model are that is is very lightweight and all of the individual parts are already well studied and efficient. This also means that the models are very lightweight, and relatively simple in comparison to more complex techniques.

The disadvantages of the model are also rooted in it's simplicity. The result of processing each image independently means that the interactions between frames is not very well represented. While the model is able to represent individual frame features very well, due to the fact that the feature maps are passed through the LSTM, classes that require specific movement from one frame to another are difficult to represent using this structure. Constructing these individual feature maps can also fall victim to background interference, meaning that a movement in the camera, or change in background could impact in a way that detracts from the main subject of the action more with this model than the other approaches discussed later in this chapter.

Long-term Recurrent Convolutional Networks [6], is a model constructed using this methodology. In the paper, they use the notation that each frame, x_i, is fed into the CNN in order to construct a fixed-length feature representation, $\phi_v(x_i)$. This is

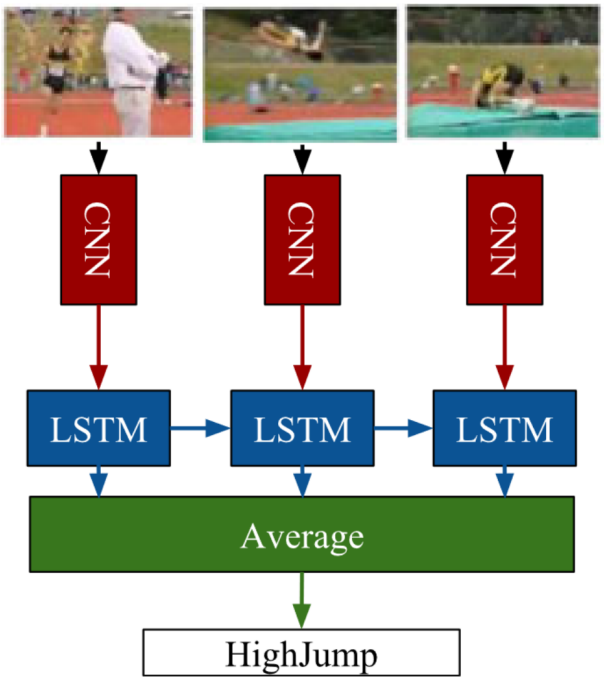

FIGURE 3.3: Action recognition structure for the LRCN model. [6]

then passed into the recurrent sequence learning model. This is where the model differs from the previous example provided. In the LRCN model, the LSTM outputs at each frame are averaged to get the output class, rather than taking the last output. This removes any bias the model may have towards the later frames in long videos. In addition to RGB frames, this model additonally uses the optical flow feature, which easily adapts to this structure, replacing the RGB frames in figure 3.3. The LSTM structure is taken from [7], which is a structure devised from the original LSTM model, as we discussed in section 2.5. The CNN's, represented in the paper as ϕ, is described as a hybrid of the CaffeNet [8] (a variant of the AlexNet [9] model discussed in section 2.3.1) and the Zeiler and Fergus [10] models, which has been pre-trained on a large dataset.

Beyond Short Snippets: Deep Networks for Video Classification [11], is another approach to this structure, which explores a more complex deep-LSTM based module, as well as more classical feature pooling. Similarly to the previously discussed model, Long-term Recurrent Convolutional Networks [6], the model utilizes a combination of two popular CNN models, AlexNet [9] and GoogLeNet [12]. The paper did explore many more classical feature pooling architectures, and were proven to have good results, however these techniques were outperformed by the

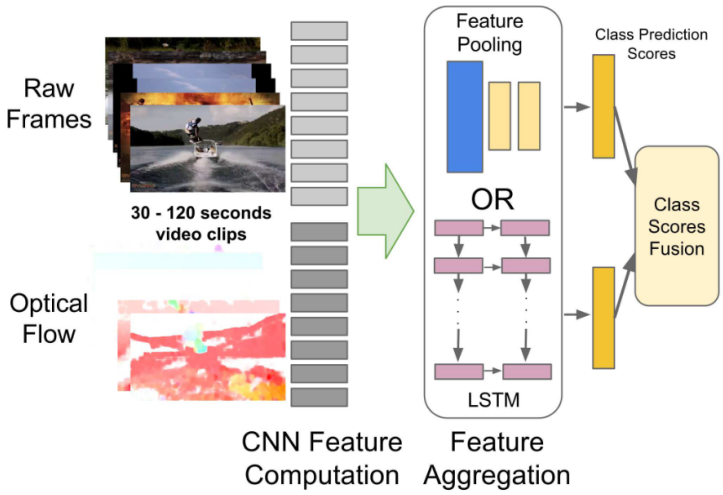

FIGURE 3.4: Overview of the Beyond Short Snippets: Deep Networks for Video Classification model [11]

LSTM model. The paper utilized a deep LSTM architecture for the feature aggregation step, shown in figure 3.5, which further adds to it's complexity, moving it above the CNN-LSTM architectures described previously. In this deep-LSTM module, the outputs of each frame are passed into a LSTM module as in the previous model, but the ouputs are then passed up through 4 more stacked layers of LSTM's, after reaching softmax layers which are averaged to get an output. These 4 additional layers of LSTM's mean it is more able to infer data moving from one frame to another. This model additionally explored the uses of optical flow and found that it adds a great deal to the accuracy of the model.

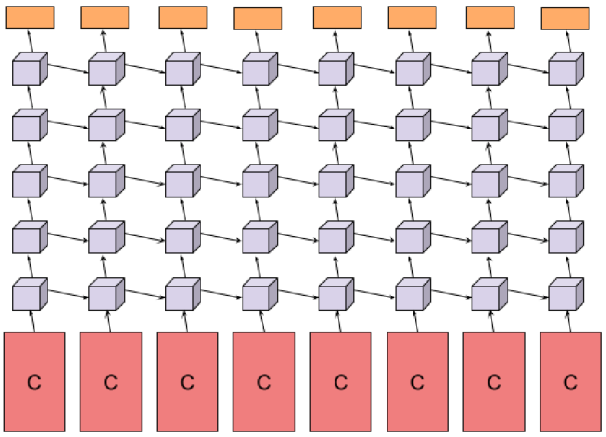

FIGURE 3.5: Deep LSTM architecture utilized by [11] in the feature aggregation step as shown in figure 3.4.

3.3.2 3D CNN Models

When considering how to handle videos without the LSTM component, the one of the first approaches that was developed is to utilize 3 dimensional CNN kernels, the specifics of which were described in section 2.2.1. The function of these kernels when it relates to action recognition is that they allow for the model to easily encode local temporal data using the third kernel dimension. The primary issue with these models is that they contain many more parameters over the 2D CNN models, meaning that they take longer to train and require more computing power as compared to the lightweight counterparts.

3D Convolutional Neural Networks for Human Action Recognition [13] was one of the original papers that proposed this model for the purposes of action recognition, and the greater topic of 3 dimensional convolutions as described in section 2.2.1. The general architecture of the model is shown in figure 3.6, and is extremely similar to that of 2 dimensional CNNs, with convolutional layers which are followed by subsampling layers.

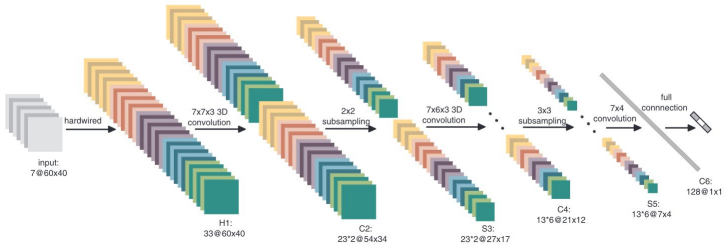

FIGURE 3.6: The original 3D-CNN action recognition model architecture proposed by [13], containing 3 convolutional layers, two subsampling layers, and one fully connected layer

The primary difference with this original architecture compared to 2D CNN's as we know them today is that it used rather large $7x7x3$ convolutions, as compared to the typical $3x3$ convolutions used in classical 2D CNN's. **Learning Spatiotemporal Features with 3D Convolutional Networks** [14] is a slightly more modern architecture that was proposed. They explore in great detail the effects of these sizes of convolutions and find that this size of convolutions are more effective than the previous methods and sizes.

Two-Stream Inflated 3D ConvNets, commonly referred to as I3D [4], is a modern variation on 3D CNN based networks. Similar to the previously discussed model

[13], this model explores the viability of taking techniques used in 2D CNN models and applying them to 3D. However it takes a much more direct approach, stating that they take the square filters of size NxN and convert them simply to 3D filters with dimensions $NxNxN$, a process they describe as *inflating* the convolutions. This inflation of convolutions allows for I3D to replicate successful 2D CNN's in their structure and apply them to video with little modifications.

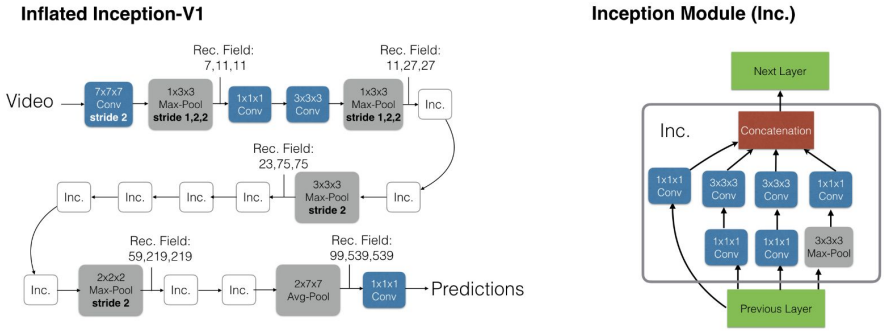

FIGURE 3.7: The model architecture used in the I3D paper [4], where the Inflated Inception-V1 architecture (left) and it's detailed submodule (right) are shown.

3.4 Model Evolution

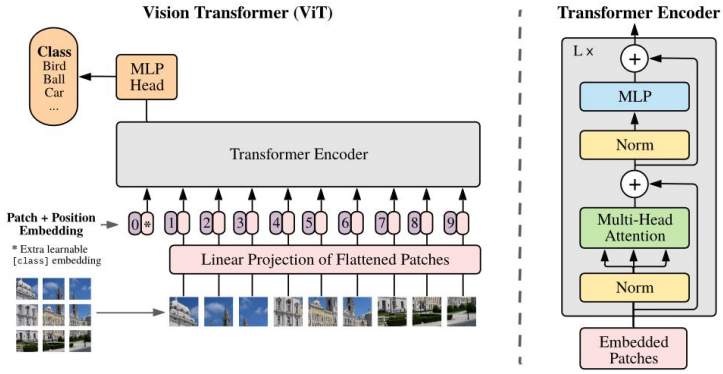

FIGURE 3.8: The original transformer model proposed in [15], the image is split into fixed-size patches, linearly embed them, and add positional embeddings. It is then fed into a standard Transformer Encoder architecture.

Transformers for Image Recognition at Scale [15], extends beyond CNN's to explore transformer networks. While transformer networks are very easily applied

to natural language processing tasks, it is not as easily applied to video and in particular action recognition. As depicted in figure 3.8, the model utilizes a standard transformer architecture in order to learn features that are useful for action recognition. The goal of this being to leverage previously well studied NLP studies that indicate the attention features of transformers are useful for focusing on the relevant data. Figure 3.9 shows this effect, the goal of this being to mitigate the challenges of handling background data interference as previously described in section 1.3. This logic is then further expanded upon in many future models to extend this functionality, such as **Multiview Transformers for Video Recognition** [16] which explores using multiple separate encoders to explore multiple views.

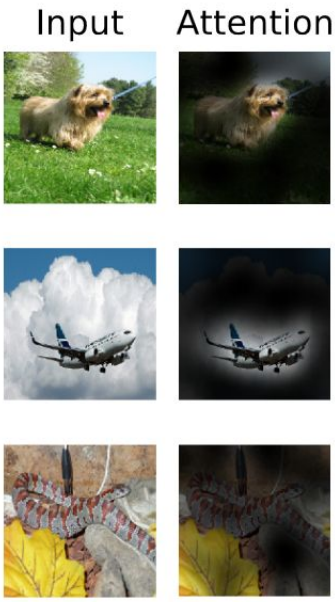

FIGURE 3.9: Example feature outputs of how a transformer utilizes attention to focus on the main subject of a video in order to greater identify actions as shown in [15]

3.5 Datasets

There are many action recognition datasets, some of which fit more specific use cases, and others which are more general and tailor to more in-the-wild data. For example the Charades dataset [17] focuses more on actions of people indoors and their interactions, whereas other datasets such as THUMOS [18] focus on many in-the-wild videos, where the location of the person can be different.

The Kinetics Human Action Video Dataset [19] is one of the primary in-the-wild action recognition datasets that are reported on by modern models. The primary advantage of this dataset over others that were published around the same time such as the UCF-101 dataset [20] is that as opposed to UCF's 101 classes and approximately 13,000 clips, the original kinetics dataset contained 400 classes and approximately 300,000 total clips which is magnitudes greater than other datasets of this type. This dataset has also been updated in 2020 to include 700 classes and over 600,000 clips. The extremely large amount of these clips, as well as containing all of; singular person actions (eg. headbanging, stretching), person-person actions (eg. handshake, tickling), and person-object actions (eg. riding a bike) among others creates a dataset that is difficult for a model to determine what action is being performed.

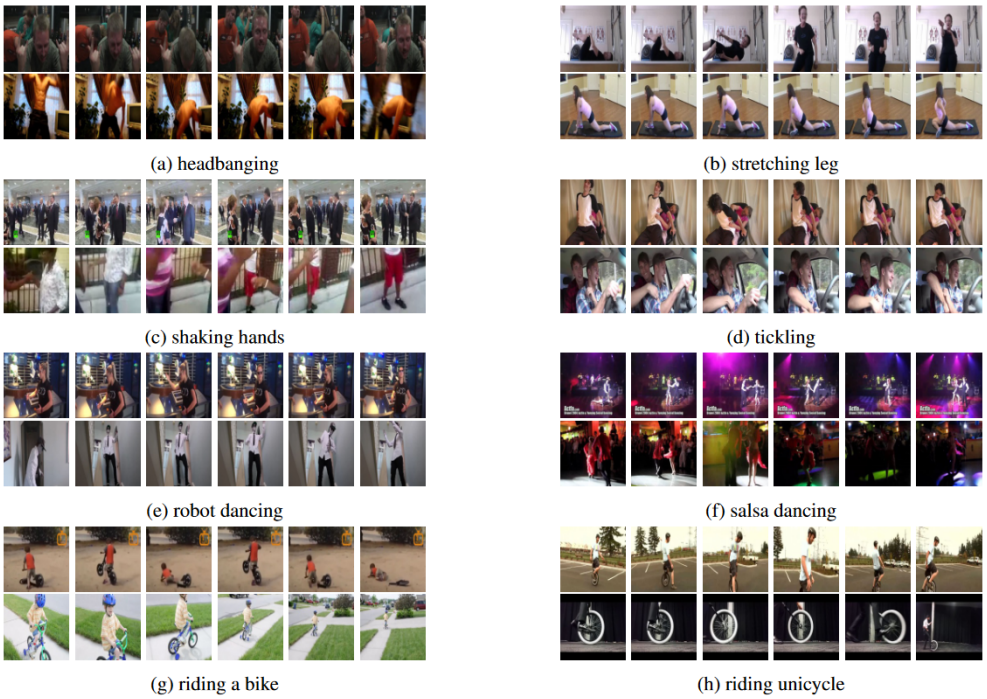

FIGURE 3.10: Example classes from the Kinetics dataset [19] which demonstrate the different singular person, person-person, and person-object interactions characterized in the dataset.

JHMDB [21] is the primary dataset that will be used in this paper. The JHMDB paper does not actually propose an entirely new dataset, rather it proposes a subset of the HMDB dataset [22], with the addition of several features. While the dataset does offer more than only annotated poses, the main appeal of the dataset when

considering the method used in this thesis is that they are fully annotated and adjusted poses to ensure that they are correct, meaning that the model can be independently be evaluated without having to consider the accuracy of the pose estimation model. Additionally, the dataset has been pruned such that the actions within the dataset only contain single person interactions, that lends itself very well to pose-based models, as generally only pose can be considered and the model can provide accurate results. Both of these features result in the dataset being highly popular with 2D-pose based models, and in particular models that create intermediate representations with these poses, as the data that is extracted from pose is both accurate and relevant to the action.

3.6 Pose Detection

Throughout the majority of this thesis, and in some pose-based papers, the concept that there is some form of extracted pose from any given RGB frame is assumed to be accurate and complete. However, the extracting of these pose features, especially in the wild, is a difficult task in and of itself. This means that when considering pose-based models (as will be done further in section 3.7) that the accuracy of these techniques must be taken into account. Without an accurate pose model, it is impossible for these pose-based action recognition models to perform with any level of accuracy. Also worth noting is that some training data can have manually annotated pose data, an example being the previously discussed JHMDB dataset [21], which utilizes manually verified pose data. A very common model utilized by pose-based models is **OpenPose** [23], which as shown in figure 3.11, is capable of detecting the poses of many people within the frame. In addition to what is shown in the figure, some models also utilize the joint heatmaps, which can also be easily generated by OpenPose. This is done through a modern technique using large CNN's and leveraging Part Affinity Fields, it is also allows for very fast real-time pose estimation.

There are many pose estimation models, as it is in itself a problem in the domain of computer vision that is constantly evolving. These can range from transformer based models such as ViTPose [24] designed to maximize dataset metrics, to the very

FIGURE 3.11: Demonstrating the effectiveness of the OpenPose [23] model. The **top** image showing that it is capable of distinguishing individual people, the **bottom left** showing the Part Affinity Fields corresponding to the limb connecting the right elbow and wrist. The **bottom right** shows a zoomed in view of these Part Affinity Fields.

lightweight models such as MoveNet developed by TensorFlow [25], developed for the purposes of real-time pose detection through mobile devices.

3.7 Pose-Based Action Recognition

Pose-based action recognition models have been well studied, and have been one of the popular forms of action recognition model as people typically determine actions by examining how a person is moving. This is because by focusing on the pose (sometimes referred to as the skeleton) of the person, you are able to effectively mitigate the background effects that were discussed in section 1.3. This means that typically more lightweight models can be used as they are able to be pointed more towards the main subject rather than filter out background data. Of course this method also comes with challenges, notably that in testing in the wild, there must be effectively two models, one to extract the pose from the person(s) in the frame, and one to process this pose data and export an action. This can introduce another point of failure, but as discussed in section 3.6, 2D pose detection has been consistently improving to the point that high quality pose data from fast models is the norm.

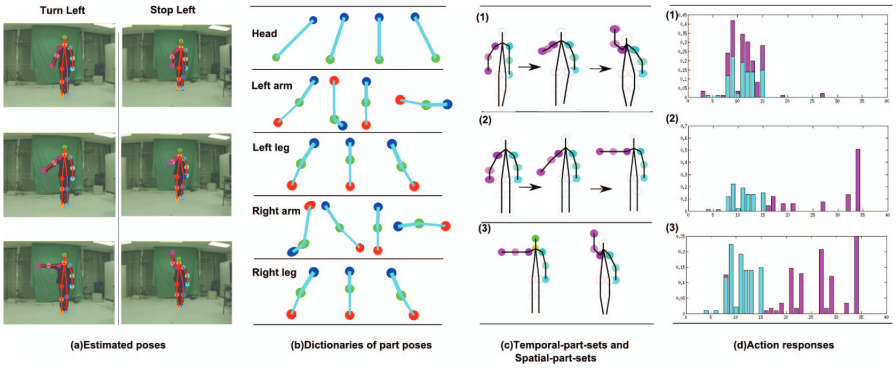

FIGURE 3.12: The overall framework of the action recognition model used by *An approach to pose-based action recognition* [26]. **(a)** & **(b)** show the esimated poses which are then used to create the dictionary of part poses. The temporal and spacial part sets in **(c)** are then represented in the histograms shown in **(d)**.

An approach to pose-based action recognition [26] was a paper that proposed a technique to pose-based action recognition that did not utilize any of the typical popular CNN's that are used in modern models. Instead this model utilizes a dictionary of part poses that is then used by a bag-of-words model, a model typically geared towards the domain of NLP. This representation is shown in figure 3.12, and while it performs well, it does suffer in that the bag-of-words models are not able to adapt to more sophisticated datasets.

Pose-Based CNN Features for Action Recognition (P-CNN) [27] is another model that utilizes the pose. Instead however, they use patches of the RGB frames centered on the various joints that have been detected, this is shown more in detail in figure 3.13. While this model does improve on typical models by using pose data, it does still struggle from the fact that it uses the raw RGB frame data, meaning that the model cannot be very small in order to handle this data.

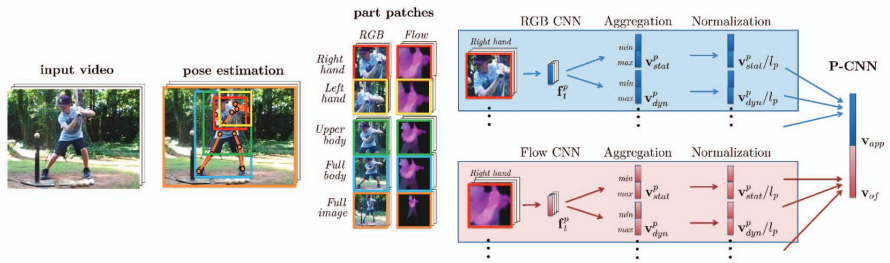

FIGURE 3.13: Illustration of P-CNN feature construction. RGB & Optical Flow "Patches" are extracted around each joint, and sometimes containing multiple joints. These features are then passed through their respective CNN's, Aggregation, & Normalization and then concatenated to form the final P-CNN feature.

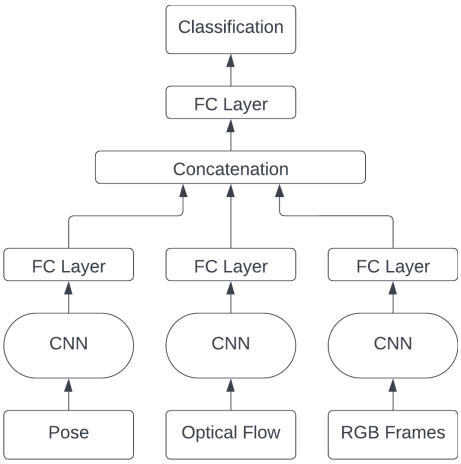

FIGURE 3.14: A typical fused architecture. Each of the Pose, Optical Flow, and RGB Frames are passed through individual 3D-CNN's, the outputs are then concatenated to achieve a final output.

Fusion-based architectures have had success in combining techniques used by basic 3D-CNN's, as seen in the I3D model [4], which as previously discussed in section 3.3.2 utilizes the RGB Frames & Optical Flow in order to predict actions. Pose can be added to this architecture as shown in figure 3.14, where the pose data is added using some additional representation, the simple being the joint heatmaps exported from a previous model, and becoming more complex with intermediate representations that will be further discussed in section 3.7.1. **Chained Multi-stream Networks Exploiting Pose, Motion, and Appearance for Action Classification and Detection** [28] is an addition to this type of model, where instead of the classic fused architecture, the model has individual loss functions added to each of the outputs, increasing performance while not adding very much additional complexity to the model.

3.7.1 Intermediate Representations

Intermediate representations are the basis for what will be discussed in later sections of this thesis. Intermediate representations aim to reduce the pose data that has been extracted into simpler and more processable formats. This is generally done with the aim of using a much more lightweight model (sometimes 2D CNN rather than 3D) that requires significantly less computing power. Of course this kind of pre-processing comes with issues, notably that by converting the model into a format such as this, some data will inevitably be lost during the transition, so the problem

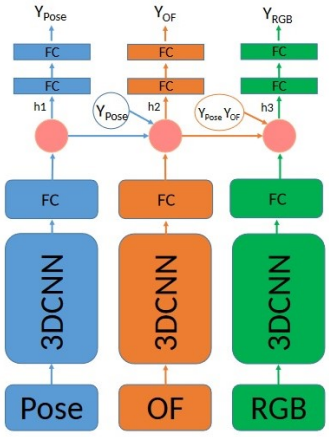

FIGURE 3.15: The chained architecture as shown in Chained Multi-stream Netowrks [28]. The model differentiates in that it has separate loss functions for each of Pose, Optical Flow, and RGB, which are chained together in a way that they can be individually optimized.

definition shifts slightly to creating an intermediate representation that both allows for a lightweight model to be effectively trained on it and for the smallest amount of data to be lost in the transition.

Pose MoTion Representation for Action Recognition (PoTion) [29] proposed one of these intermediate representations, however they did it in a way that was unique in that they only considered the joint positions rather than the skeletons themselves. Namely, the model utilizes the joint positions of a person throughout each frame of video to construct 2 dimensional images that reflect the movement of each of these joints. This is done by stacking each joint heatmap onto one image, and colorize them according to the point in time the frame is extracted. This colorization technique is shown in more detail in figure 3.17. The overall representation construction is shown in figure 3.16, and shows how after the colourization is performed, the heatmap images are then stacked together. These stacked images can then be passed into a simple 2D CNN, which can be quickly and efficiently trained and performs rather well. In the paper they also explore adding this implementation as another input to the I3D [4] model in conjunction with the optical flow and rgb frames, which it showed to offer an increase in performance to the existing model. **Pose and Joint-Aware Action Recognition** [30] is a slight improvement on this model structure, they utilize a similar colorization scheme. Instead of feeding all of the joints into the model, they developed a joint-motion re-weighting network

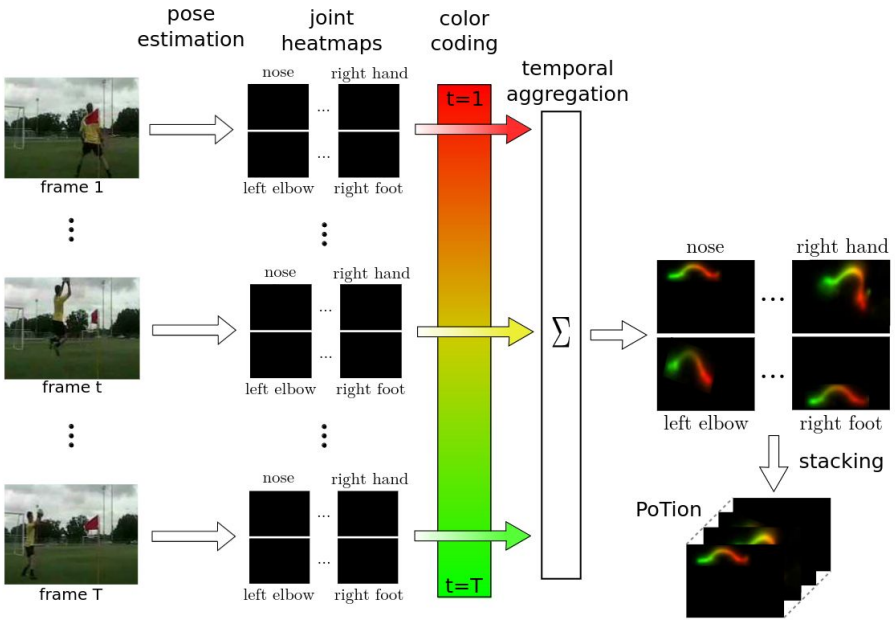

FIGURE 3.16: The illustration of the PoTion representation [29]. The input joint heatmaps are colored based on their time in the frame, and the frames are then concatenated to form the final movement of the joint throughout the video.

(denoted in the paper as JMRN), which allowed for the model to easily find the dependencies between joints. This joint selection procedure allowed for the model to offer improved performance over the original PoTion model.

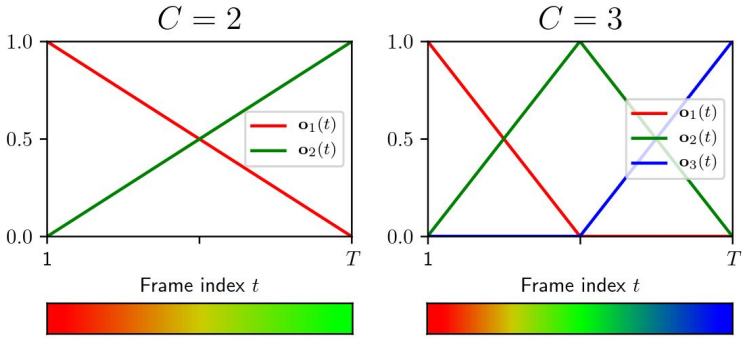

FIGURE 3.17: The colourization method utilized by the PoTion model [29]. As the frame index moves throughout the video, the colour of the joint shifts from one to another. This can be done for any amount of colours, denoted by C, the figure shows examples for C=2 and C=3, but the same logic holds for more than 3.

Pose-Action 3D Machine for Video Recognition (PA3D) [31] takes a similar approach to the PoTion model, but flavours it a bit differently. Similar to PoTion model, it leverages joint heatmaps similar to that exported from the OpenPose pose detection model, however it does not colorize the joints and aim to insert them into one

image. A part of the model known as the *Temporal Pose Convolution*, shown in figure 3.18 is a core part of how the model functions. This is done through 1x1xN convolutions, which are run stacks of the joint heatmap images.

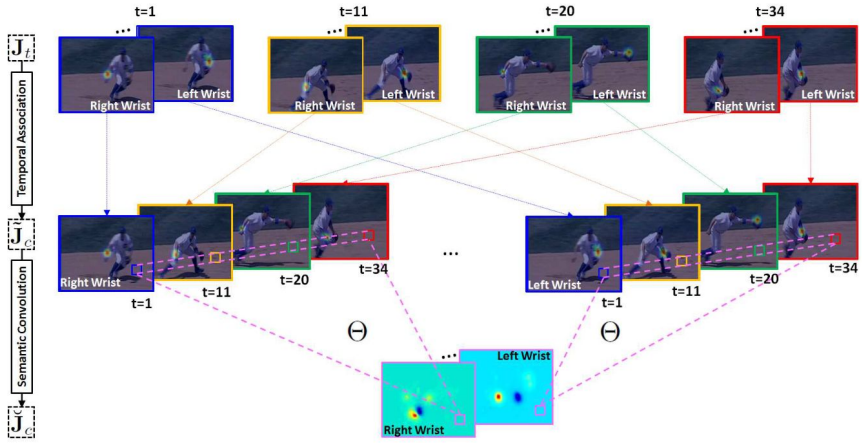

FIGURE 3.18: The main PA3D [31] model architecture, demonstrating the 1x1 convolutions used in order to construct the temporal cube.

Simple yet efficient real-time pose-based action recognition [32] was largely the inspiration for work done on this thesis. The goal of this paper was to provide a very lightweight and simple to understand intermediate representation that could be used by a very simple CNN to perform real-time action recognition. They do this by converting the skeletons into their unique *Encoded Human Pose Image (EHPI)*, which is fundamentally just a 2d grid where the x-axis is frame index, and the y axis is the joint, this is shown in figure 3.19. This EHPI representation can then be used with a very simple CNN to provide very fast and good results in order to process actions in real time. There is one notable disadvantage in that it relies so heavily on the global positioning of the person in the frame. This means that the representation is very sensitive to things such as camera movement, where a slight movement results in the representation interpreting as the whole person sliding throughout the frame, however this could be mitigated via person detection to keep the person centered in the frame.

Make Skeleton-based Action Recognition Model Smaller, Faster and Better [33] is yet another improvement on this intermediate representations, but with the particular focus on making the representations more resistant to both rotation & shifting of a person throughout the frame. As shown in figure 3.20, this is done

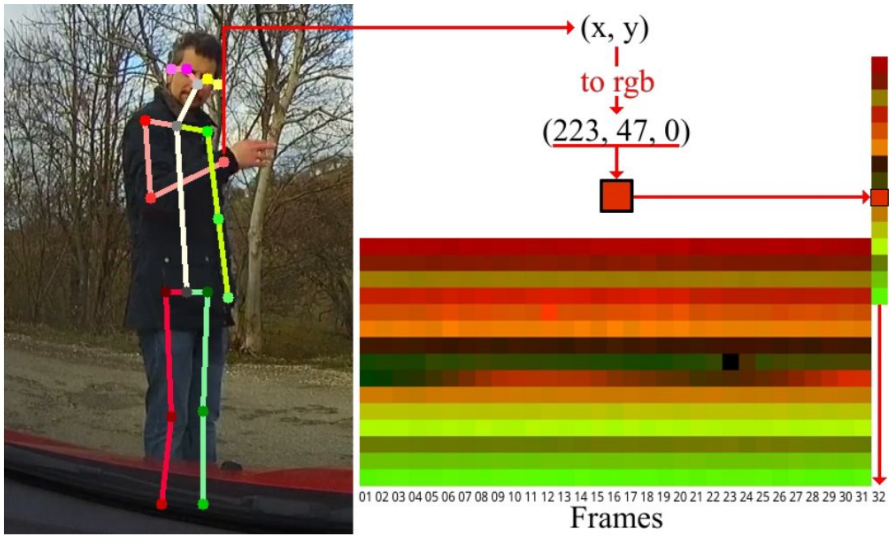

FIGURE 3.19: The EHPI representation used in the Simple yet efficient paper [32]. The x, y coordinates of each joint are mapped to the red & green values of a pixel, all joints are then stacked to form a column of joint positions in a frame. Each frame is then placed next to each other to form the 2D representation.

through two particular features, the cartesian coordinate feature which was used in previous models in a similar way [32], however it is done slightly differently through the JCD feature. The JCD feature is indifferent to rotations and shifting since all of the representation is aware of is the distance between any two given joints. This allows for easier generalization, however in the final model, both the cartesian coordinate and JCD features are used, as the authors determined that both were key in achieving high performance.

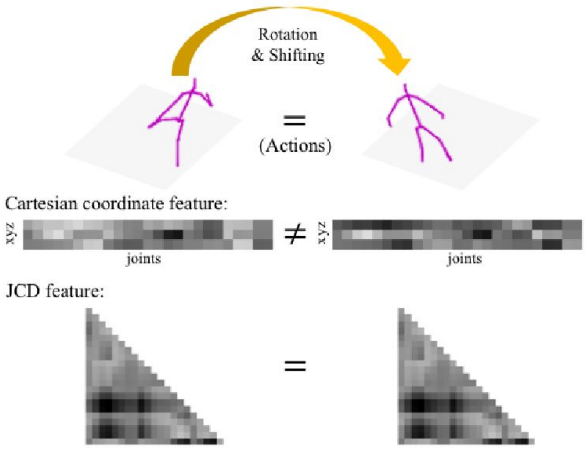

FIGURE 3.20: The representation used by the Smaller, Faster, Better model [33], this is split into two representations. The cartesian coordinates of each joint are encoded into a 2d representation, not dissimilar to previously discussed models [32]. The JCD feature is a similar representation, but instead of x, y, and z coordinates, uses the distance between two joints.

Chapter 4

Methodology

Chapter 5

Experimentation

Chapter 6

Conclusion

Appendix A

Frequently Asked Questions

A.1 How do I change the colors of links?

The color of links can be changed to your liking using:

`\hypersetup{urlcolor=red}`, or

`\hypersetup{citecolor=green}`, or

`\hypersetup{allcolor=blue}`.

If you want to completely hide the links, you can use:

`\hypersetup{allcolors=.}`, or even better:

`\hypersetup{hidelinks}`.

If you want to have obvious links in the PDF but not the printed text, use:

`\hypersetup{colorlinks=false}`.

Bibliography

[1] H. Sun and Y. Chen, *Real-time elderly monitoring for senior safety by lightweight human action recognition*, 2022. arXiv: 2207.10519 [cs.CV].

[2] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[3] M. Tan and Q. V. Le, *Efficientnetv2: Smaller models and faster training*, 2021. arXiv: 2104.00298 [cs.CV].

[4] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[5] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime tv-l1 optical flow," in *Pattern Recognition*, F. A. Hamprecht, C. Schnörr, and B. Jähne, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 214–223, ISBN: 978-3-540-74936-3.

[6] J. Donahue, L. A. Hendricks, M. Rohrbach, *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 677–691, 2017. DOI: 10.1109/TPAMI.2016.2599174.

[7] W. Zaremba and I. Sutskever, *Learning to execute*, 2015. arXiv: 1410.4615 [cs.NE].

[8] Z. Qin, F. Yu, C. Liu, and X. Chen, "How convolutional neural networks see the world — a survey of convolutional neural network visualization methods," *Mathematical Foundations of Computing*, vol. 1, pp. 149–180, Jan. 2018. DOI: 10.3934/mfc.2018008.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25, Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[10] M. D. Zeiler and R. Fergus, *Visualizing and understanding convolutional networks*, 2013. arXiv: 1311.2901 [cs.CV].

[11] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4694–4702. DOI: 10.1109/CVPR.2015.7299101.

[12] C. Szegedy, W. Liu, Y. Jia, *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[13] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013. DOI: 10.1109/TPAMI.2012.59.

[14] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, *Learning spatiotemporal features with 3d convolutional networks*, 2015. arXiv: 1412.0767 [cs.CV].

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, *An image is worth 16x16 words: Transformers for image recognition at scale*, 2021. arXiv: 2010.11929 [cs.CV].

[16] S. Yan, X. Xiong, A. Arnab, *et al.*, "Multiview transformers for video recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 3333–3343.

[17] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, *Hollywood in homes: Crowdsourcing data collection for activity understanding*, 2016. arXiv: 1604.01753 [cs.CV].

[18] A. Gorban, H. Idrees, Y.-G. Jiang, *et al.*, *THUMOS challenge: Action recognition with a large number of classes*, http://www.thumos.info/, 2015.

[19] W. Kay, J. Carreira, K. Simonyan, *et al.*, *The kinetics human action video dataset*, 2017. arXiv: 1705.06950 [cs.CV].

[20] K. Soomro, A. R. Zamir, and M. Shah, *Ucf101: A dataset of 101 human actions classes from videos in the wild*, 2012. arXiv: 1212.0402 [cs.CV].

[21] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, "Towards understanding action recognition," in *International Conf. on Computer Vision (ICCV)*, Dec. 2013, pp. 3192–3199.

[22] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: A large video database for human motion recognition," in *2011 International conference on computer vision*, IEEE, 2011, pp. 2556–2563.

[23] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7291–7299.

[24] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, *Vitpose: Simple vision transformer baselines for human pose estimation*, 2022. arXiv: 2204.12484 [cs.CV].

[25] M. Abadi, A. Agarwal, P. Barham, *et al.*, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015. [Online]. Available: https://www.tensorflow.org/.

[26] C. Wang, Y. Wang, and A. L. Yuille, "An approach to pose-based action recognition," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 915–922. DOI: 10.1109/CVPR.2013.123.

[27] G. Chéron, I. Laptev, and C. Schmid, "P-cnn: Pose-based cnn features for action recognition," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3218–3226. DOI: 10.1109/ICCV.2015.368.

[28] M. Zolfaghari, G. L. Oliveira, N. Sedaghat, and T. Brox, "Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[29] V. Choutas, P. Weinzaepfel, J. Revaud, and C. Schmid, "Potion: Pose motion representation for action recognition," Jun. 2018, pp. 7024–7033. DOI: 10.1109/CVPR.2018.00734.

[30] A. Shah, S. Mishra, A. Bansal, J.-C. Chen, R. Chellappa, and A. Shrivastava, "Pose and joint-aware action recognition," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2022, pp. 3850–3860.

[31] A. Yan, Y. Wang, Z. Li, and Y. Qiao, "Pa3d: Pose-action 3d machine for video recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7914–7923. DOI: 10.1109/CVPR.2019.00811.

[32] D. Ludl, T. Gulde, and C. Curio, "Simple yet efficient real-time pose-based action recognition," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 581–588. DOI: 10.1109/ITSC.2019.8917128.

[33] F. Yang, Y. Wu, S. Sakti, and S. Nakamura, "Make skeleton-based action recognition model smaller, faster and better," in *Proceedings of the ACM Multimedia Asia*, ser. MMAsia '19, Beijing, China: Association for Computing Machinery, 2020, ISBN: 9781450368414. DOI: 10.1145/3338533.3366569. [Online]. Available: https://doi.org/10.1145/3338533.3366569.