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

June 25, 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

1.5 Thesis Structure

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

3.2 Optical Flow

3.3 CNN Based Models

Naturally, the first approach to feeding video data into models is to process the raw RGB frames. The RGB frame data quite often

3.3.1 CNN + LSTM Models

The success of CNN's in the world of image classification makes the move to apply the same type of logic towards action recognition and the larger domain of video processing. Since a video broken down is just individual frames, the logic follows that we would be able to extract features from individual frames and combine these features to produce a classification outcome.

In very classic models, this 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 simply 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.

Figure 3.1 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.

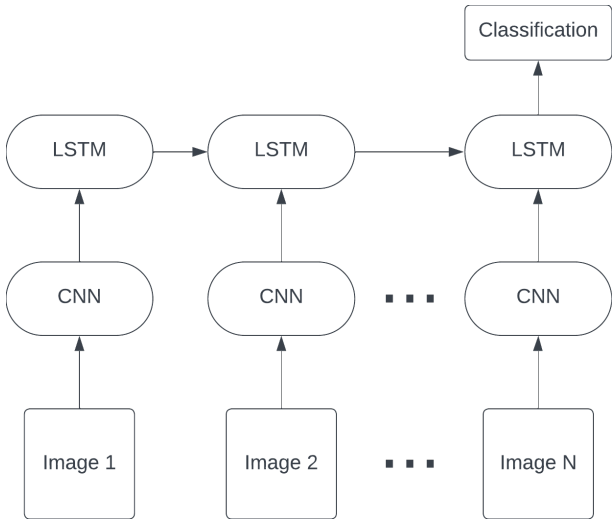

FIGURE 3.1: An example structure of a simple CNN-LSTM based model, each individual frame being individually fed into the CNN, and then passed to a LSTM.

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 [2], 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 then passed into the recurrent sequence learning model. This is where the model

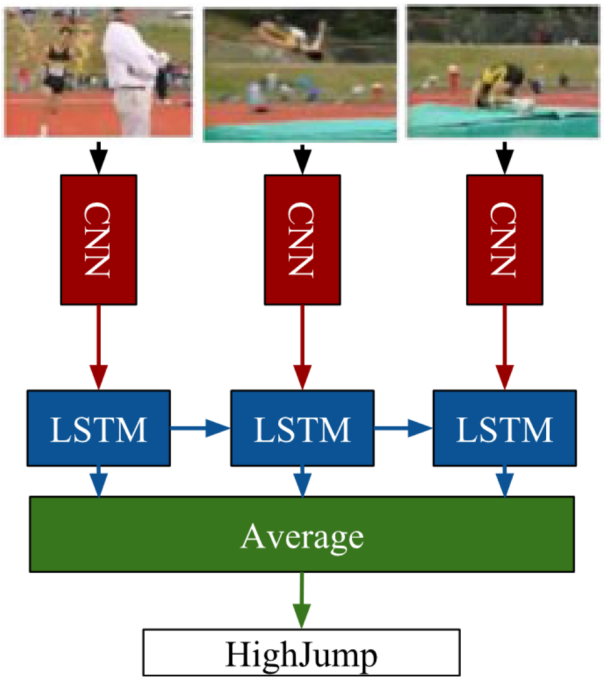

FIGURE 3.2: Action recognition structure for the LRCN model. [2]

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 3.2. The LSTM structure is taken from [3], 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 [4] (a variant of the AlexNet [5] model discussed in section 2.3.1) and the Zeiler and Fergus [6] models, which has been pre-trained on a large dataset.

Beyond Short Snippets: Deep Networks for Video Classification [7], 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 [2], the model utilizes a combination of two popular CNN models, AlexNet [5] and GoogLeNet [8]. The paper did explore many more classical feature pooling architectures, and were proven to have good results, however these techniques were outperformed by the LSTM model. The paper utilized a deep LSTM architecture for the feature aggregation

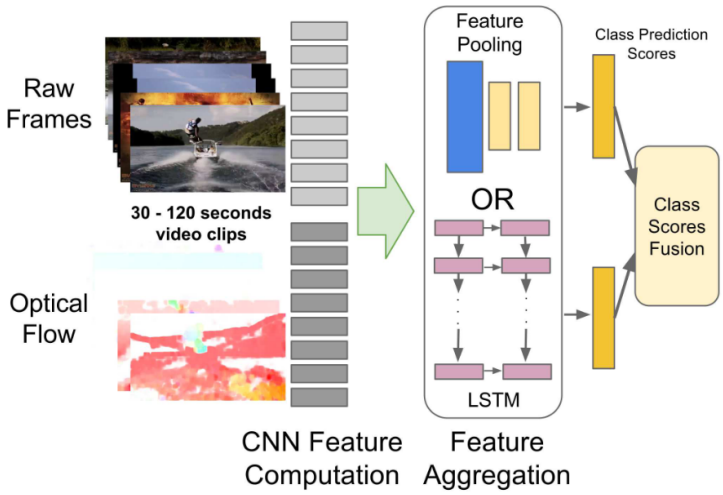

FIGURE 3.3: Overview of the Beyond Short Snippets: Deep Networks for Video Classification model [7]

step, shown in figure 3.4, 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.

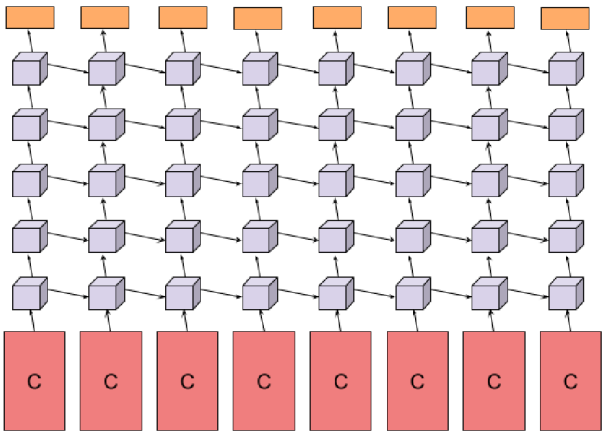

FIGURE 3.4: Deep LSTM architecture utilized by [7] in the feature aggregation step as shown in figure 3.3.

3.3.2 3D CNN Models

When considering how to handle videos without the LSTM component, the most natural approach 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 [9] was one of the original paper 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 3.5, and is very very similar to that of 2 dimensional CNNs, with convolutional layers followed up by subsampling layers.

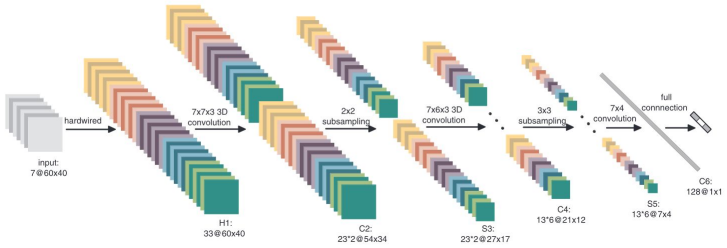

FIGURE 3.5: The original 3D-CNN action recognition model architecture proposed by [9], containing 3 convolutional layers, two subsampling layers, and one fully connected layer

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** [10], 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 [13], is a modern variation on 3D CNN based networks. Similar to the previously discussed model

[9], 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.

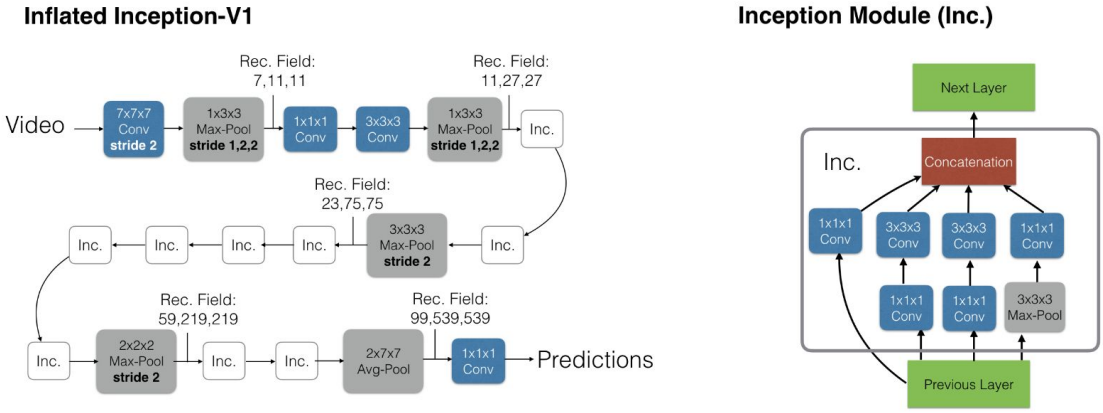

FIGURE 3.6: The model architecture used in the I3D paper [13], where the Inflated Inception-V1 architecture (left) and it's detailed submodule (right) are shown.

3.4 Model Evolution

The domain of action recognition is always evolving, and it would not be a complete review of the literature without acknowledging other approaches that extend beyond the reach of this thesis.

Transformers for Image Recognition at Scale [11], 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 3.7, the model utilizes a standard transformer architecture often used in NLP tasks 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 3.8 shows this effect, the goal of this being to mitigate the challenges of handling background data interference as previously described in

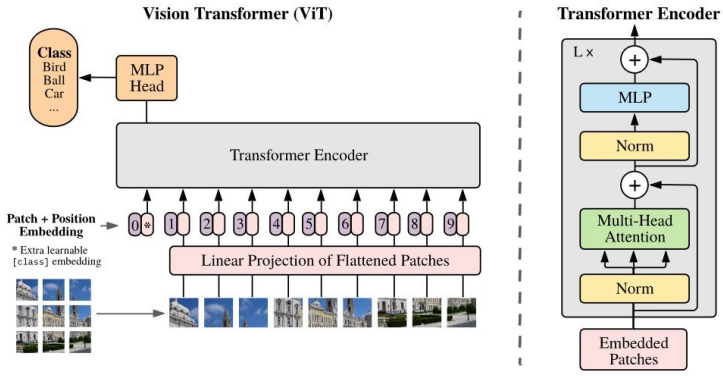

FIGURE 3.7: The original transformer model proposed in [11], the image is split into fixed-size patches, linearly embed them, and add positional embeddings. It is then fed into a standard Transformer Encoder architecture.

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** [12] which explores using multiple separate encoders to explore multiple views.

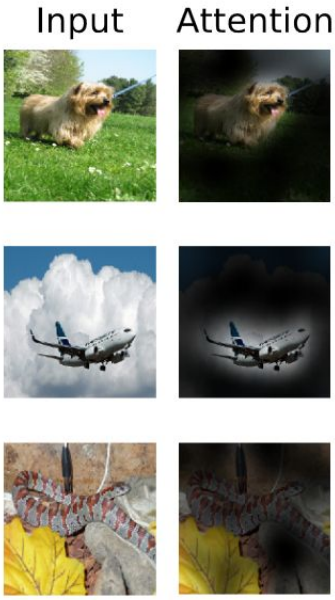

FIGURE 3.8: Example feature outputs of how a transformer utilizes attention to focus on the main subject of a video in order to greater identify actions as shown in [11]

3.5 Datasets

3.5.1 Kinetics

3.5.2 JHMDB

3.6 Skeleton-Based Action Recognition

3.6.1 Pose Detection

3.6.2 Intermediate Representations

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] 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.

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

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

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

[11] 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].

[12] 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.

[13] 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.