IN SEARCH OF VIDEO TRANSFORMER MODELS FOR ACTION RECOGNITION ON LIMITED DATA

JEGOR KITŠKERKIN

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF SCIENCE IN COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE

DEPARTMENT OF
COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

STUDENT NUMBER

2041771

COMMITTEE

dr. Sharon Ong
prof. dr. Eric Postma

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

May 20, 2022

ACKNOWLEDGMENTS

IN SEARCH OF VIDEO TRANSFORMER MODELS FOR ACTION RECOGNITION ON LIMITED DATA

JEGOR KITŠKERKIN

Abstract

Video understanding is an inherently complex problem. Although rich in informational content, videos are hard to tackle due to the presence of both spatial and temporal axes, while at the same time being naturally large in size. Famous for their requirement for the immensely large dataset sizes, the Transformer architecture has shown to be a success in the domain of natural language processing, however the current research in other domains is still on-going. This study tries to find an answer on whether it is possible to use Transformer models for action recognition on videos when the data size is limited. A novel approach of combining video transformer models together with keyframe selection algorithms is proposed and explored. The study tries to combine and integrate different existing regularization methods and create a multi-stage pipeline. The proposed solution is compared to the convolutional models that are more traditional for the visual domain. The final pipeline has shown to achieve 95% accuracy on the UCF-101 dataset with 101 label and just 9500 training samples. Additionally, it was found that the correct frame sampling can greatly improve the predictive power of the underlying classifier model.

Work on this thesis did not involve collecting data from human participants or animals. The original owner of the data, the weights of the pretrained models and code used in this thesis retains ownership of the data and code during and after the completion of this thesis. The author of this thesis acknowledges that they do not have any legal claim to this data.

1 INTRODUCTION

The overarching goal of this thesis is to find a Machine Learning pipeline for action recognition that would use a transformer-based model as its

classifier, while being an effective predictor and having the ability to be applied on small datasets without the need of industry-level computational resources.

Since 2017, the attention mechanism (Vaswani et al., 2017) has revolutionized natural language processing (NLP). Creation of BERT (Devlin, Chang, Lee, & Toutanova, 2019) has irrevocably shaped the future of NLP. Mordor Intelligence, in their 2021 report (Mordor Intelligence, 2021), forecast the NLP market to reach $48.5 billion by 2026, increasing by 27% annually. Subsequently, there has been a number of attempts to apply attention and transformer architecture to other domains, such as audio and images. Pure transformer-based vision models have achieved state-of-the-art (SOTA) results, overcoming convolution-based models that have dominated the vision domain (Arnab et al., 2021; Girdhar et al., 2022; Li et al., 2022; Liu, Ning, et al., 2021; Wang et al., 2021).

The use of transformers for action recognition is not only practically relevant, but is also interesting from the scientific point of view. Although in the recent time, there has been a number of transformer models proposed for images, the domain of videos is still greatly overlooked. There is a natural correlation between images and videos, as they both convey information through the visual channel. However, the presence of a temporal axis in addition to spatial axis present in images, adds an important source of information and complexity.

Currently, most of the SOTA results in video classification are achieved on large computational clusters, which in a lot of times heavily limits its usability for smaller business. Finding an efficient way to utilize transformers for video classification might democratize this field to many new participants.

Video information allows us to detect activities and actions. Classification of video segments can drive innovation in many sectors around the globe. The possible applications of video classification are security cameras which power automatic surveillance. Another possible usage is in autonomous vehicles, where video classification can be applied as part of the larger computer vision system, for example for traffic sign classification. Additionally, video hosting services like YouTube utilize video classification for recommendation systems and video tagging.

Human action recognition is one of the subdomains of video classification. Human action recognition involves correct classification of actions performed by humans on images or videos. Using videos for action recognition is more challenging due to the computational overload, but is also more rich in the information it provides to make a correct prediction.

Action recognition is applied in a wide range of domains and settings. The sports industry uses action recognition to facilitate correct execution of

exercises; recently a number of companies have made successful attempts to automatize the shopping industry by creating autonomous shops that use computer vision, namely action recognition paired with object detection to let customers shop without the need of traditional cashiers. Another prospective area in which human action recognition could be found useful is in elderly care homes, where there is often a lack of staff and sometimes additional guidance and observation is needed for the elderly people.

Although an enormous amount of data, including video data, is produced by the Internet every moment, it is hard to extract knowledge from this data due to the lack of labelling. Labelling datasets is a time-taking exercise. Currently, the number of video datasets publicly available compared to other domains is small. It is even harder to create large datasets for domain-specific tasks, as experts should be involved and the amount of data can be limited. Therefore, this thesis aims to address the following research question: *"To what extent can transformer-based models be used for activity recognition on small datasets?"* This research question is tackled by utilizing different regularization methods, leveraging large pretrained models and using a multi-stage model setup.

Dosovitskiy et al. (2021) mention that transformers lack some of the inductive biases that are present in traditional CNN models, and thus successful training of SOTA vision transformer models requires tens of millions of training samples. Arnab et al. (2021) observed that the largest video datasets can be orders of magnitude smaller than their image counterparts. So utilization of already existing knowledge in Image Transformers, that can be trained much more effectively, can help greatly reduce the cost of training Video Transformers and at the same time increase their accuracy. Consequently, it means that it is important to answer the subquestion *"To what extent can image transformers be used to improve classification of videos?"* This question is addressed by using the weights of a large pretrained Image Transformer model that is further fine-tuned on a video dataset.

The fact that videos are sequences of individual frames raises an important question of how to sample those frames effectively. Many videos in current day datasets are of different length, many can be too large to process (e.g. YouTube videos). Selecting only the most relevant, most informative and distinctive frames gives the ability to greatly reduce the cost of training a model. This means that it is important to see *"To what extent selection of key frames affects performance of video classification"*. It is answered by using different keyframe selection methods that are used to sample frames before they are given as input to the model.

The main finding of this thesis is that transformer-based architectures can be an effective predictive backbone for video classification, however due to the novelty of this architecture in the video domain, some models are still

computationally heavy. Nevertheless, when the right architecture is picked, and additional steps are taken to ensure powerful regularization constraints, then it is indeed possible to have a powerful yet computationally effective transformer-based pipeline. Additionally, thorough attention should be devoted to the video frame sampling so to guarantee that only the most descriptive and informationally rich frames are used to make a prediction.

2 RELATED WORK

Since the creation of LeNet (LeCun et al., 1989), visual understanding domain was dominated by convolutional neural networks.

Convolutional neural networks remained de facto standard approach for working with images and videos. The usual approach for action recognition in videos is based on two neural network types: convolutional neural networks and/or recurrent neural networks.

The first approach for modelling videos consists of pairing a large convolutional neural network that was trained on an image dataset and extending it by combining with a recurrent neural network. This way the recurrent neural network will utilize the frame level features extracted from the convolutional neural network (Zha, Luisier, Andrews, Srivastava, & Salakhutdinov, 2015). Each frame is being first processed by the convolutional neural network and the features extracted by the convolutional neural network are then fed into the recurrent layer which can model the temporal information in videos. Traditional recurrent neural networks have been shown to have a vanishing gradient problem, which often leads to poor performance over long sequences as the model tends to forget inputs at previous timesteps. To combat this, long-short term memory networks (LSTMs) were proposed (Hochreiter & Schmidhuber, 1997).

As a successor to LSTMs, Cho, van Merrienboer, Bahdanau, and Bengio (2014) proposed Gated Recurrent Unit (GRU) that uses one less gate (the memory unit) than LSTM, making it more computationally effective while still being comparable in performance, and helping converge faster on smaller datasets (Chung, Gulcehre, Cho, & Bengio, 2014).

Simonyan and Zisserman (2014) introduce two-stream networks, taking inspiration from the visual cortex system of humans, where the ventral pathway acts as spatial recognition stream and dorsal pathway acts as temporal recognition stream. Authors model the spatial and temporal stream of videos separately using convolutional networks and the softmax output of both streams are combined by late fusion either by averaging or a linear support vector machine (SVM). As the input to the temporal stream CNN, authors use stacked optical flow so that the CNN does not need to estimate the motion in frames itself and converge faster. To further

increase the overall performance, authors mention that the pre-training of the spatial CNN can be performed on a large-scale image classification dataset, as the spatial CNN is merely a regular image CNN. An interesting finding is that training is performed with two classification heads, where one is used for the UCF-101 (Soomro, Zamir, & Shah, 2012) dataset and the other for the HMDB (Kuehne, Jhuang, Garrote, Poggio, & Serre, 2011) dataset classification. Authors note that this gives possibility of utilizing additional data without the need of merging them. The two-stream model with multi-task training and late fusion by SVM achieved state-of-the-art 88% accuracy on UCF-101 and HMDB by a large margin over other existing at that moment deep networks. However, although deemed successful, the mentioned setup heavily relies on the information extracted by the optical flow estimation model, which might not be an optimal solution in terms of the predictive power of the final model. Additionally, using averaging and SVMs is limiting compared to the modern day deep neural networks.

Another architecture for video modelling is purely convolutional. This approach became possible due to the creation of larger video datasets like Kinetics 400 (Kay et al., 2017) and Sports-1M (Karpathy et al., 2014) that contain orders of magnitude bigger amount of samples than the datasets used by Simonyan and Zisserman (2014). Instead of using recurrent layers or optical flow estimation, temporal axis is modelled the same way as spatial, by means of convolution. The growth of the video datasets meant that estimating the temporal axis became possible without the need of intermediate steps like optical flow estimation. Tran et al. (2018) explore a number of possible ways to compute spatio-temporal convolution, mainly contrasting between computing the 3D convolution, where the convolution is performed along spatial and temporal axes jointly, or computing (2+1)D convolution, where the spatial and temporal convolutions are computed separately. Authors show that (2+1)D based architectures generally outperform 3D convolutions, and at the same time are less computationally intensive.

However, the success of pure attention-based models in the area of natural language understanding lead to attention mechanism being spread to other fields of machine learning. The most popular implementation of attention, that utilizes multihead self-attention has appeared in Vaswani et al. (2017) and has been called the Transformer, which lead to a whole class of new models in machine learning.

Dosovitskiy et al. (2021) present a first big implementation of the Transformer architecture in image classification, managing to achieve results comparable to convolutional neural networks, which were considered a gold standard in Computer Vision. The Vision Transformer has been made possible due to introduction of what has been called patch embed-

dings. Transformer, unlike the convolutional network, does not accept grid-like data and rather accepts one-dimensional data. Through the use of patch embeddings the image is split into non-overlapping patches and then treated as a sequence. Shortly after, a number of extensions and continuations to Vision Transformer were proposed.

VIDEO VISION TRANSFORMER: VIT FOR VIDEOS Video Vision Transformer (ViViT) explore the possible ways of computing self-attention in videos (Arnab et al., 2021). Presence of temporal axis in videos in comparison to images means that there is additional overload when computing attention. A logical extension of "patch embeddings" from ViT are "tubelet embeddings", which compute the tokens along three dimensions.

The authors proposed a number of possible ways to add temporal axis to existing Vision Transformer architecture. The authors explored four ways of how the spatio-temporal attention can be computed:

- Spatio-temporal attention
- Factorised encoder
- Factorised self-attention
- Factorised dot-product attention

It was found that a factorised encoder model had the highest performance. It modelled the temporal interaction separately after modelling the spatial interaction. The joint spatio-temporal attention performed slightly worse and had a higher computational overhead due to the higher number of parameters.

Among other things, an approach to finetune a ViViT model based on a ViT model weights is provided. Authors present a new way of inflating the embeddings weights from two dimensions to three dimensions, which they call "central frame initialisation" and has a better performance than traditional "filter inflation". This methodology gives ability to initialize a ViViT model weights from larger image models like ViT (Dosovitskiy et al., 2021) or DeIT (Touvron et al., 2021), thus leverage their existing predictive power and decreasing needed training time marginally.

However, authors note that successful training of the ViViT still requires immense amounts of data, and in case of joint spatio-temporal attention model, the computational complexity creates a large shortcoming.

HIERARCHICAL MODELLING USING SHIFTED WINDOWS Liu, Lin, et al. (2021) developed a Swin Transformer, which is a hierarchical model, similar to the popular CNN-based architectures using Shifted Windows to

address some of the complications, such as scale variations, that arise when switching the domain of transformers from language to vision. The model is built in a pyramid fashion, constructing a hierarchy of patches of different sizes from smallest to larger. Not only does this approach help account for the specifics of the vision modality, but it facilitates computational complexity, decreasing it from quadratic to linear with regards to the input size. The successful performance of Swin Transformer can be attributed to its ability for translation invariance, locality inductive bias and hierarchical modelling structure - important properties that might drive the acceptance of attention-based models as general backbone for computer vision tasks.

The Video Swin Transformer (Liu, Ning, et al., 2021), being a natural extension of Swin Transformer to videos, managed to both improve accuracy and decrease computational complexity of ViViT architecture. As its image alternative, it utilizes spatio-temporal locality bias instead of computing complex global self-attention. The authors also studied the effect of computing the temporal self-attention separately from spatial self-attention. In contrast to findings in Arnab et al. (2021), joint temporal-spatial self-attention performed the best, at the same time being least computationally expensive. Additionally, it was found that a smaller learning rate of the base model, in comparison to the learning rate of the classification head has positive impact on accuracy.

Based on the mentioned advantages of Video Swin, this architecture can be seen as the most advanced approach to video modelling using transformers, because it overcomes many of the shortcomings present in ViViT.

3 METHODS

3.1 *Experimental Setup*

As can be seen from the previous work on action recognition, the video domain is not only challenging from the perspective of computational power but also because of the heterogeneity of the problem. This poses a need for a multi-stage setup, where the different problems that arise when approaching action recognition in videos are tackled appropriately.

First and foremost problem is the irregularity of the video lengths and the video duration itself. It is still impossible to feed modern transformer models with even a single five-second video on the most recent NVIDIA A100 GPU. This complexity is tackled by using keyframe selection methods that help summarize lengthy videos in a concise manner, resulting in smaller input data while at the same time increasing the performance.

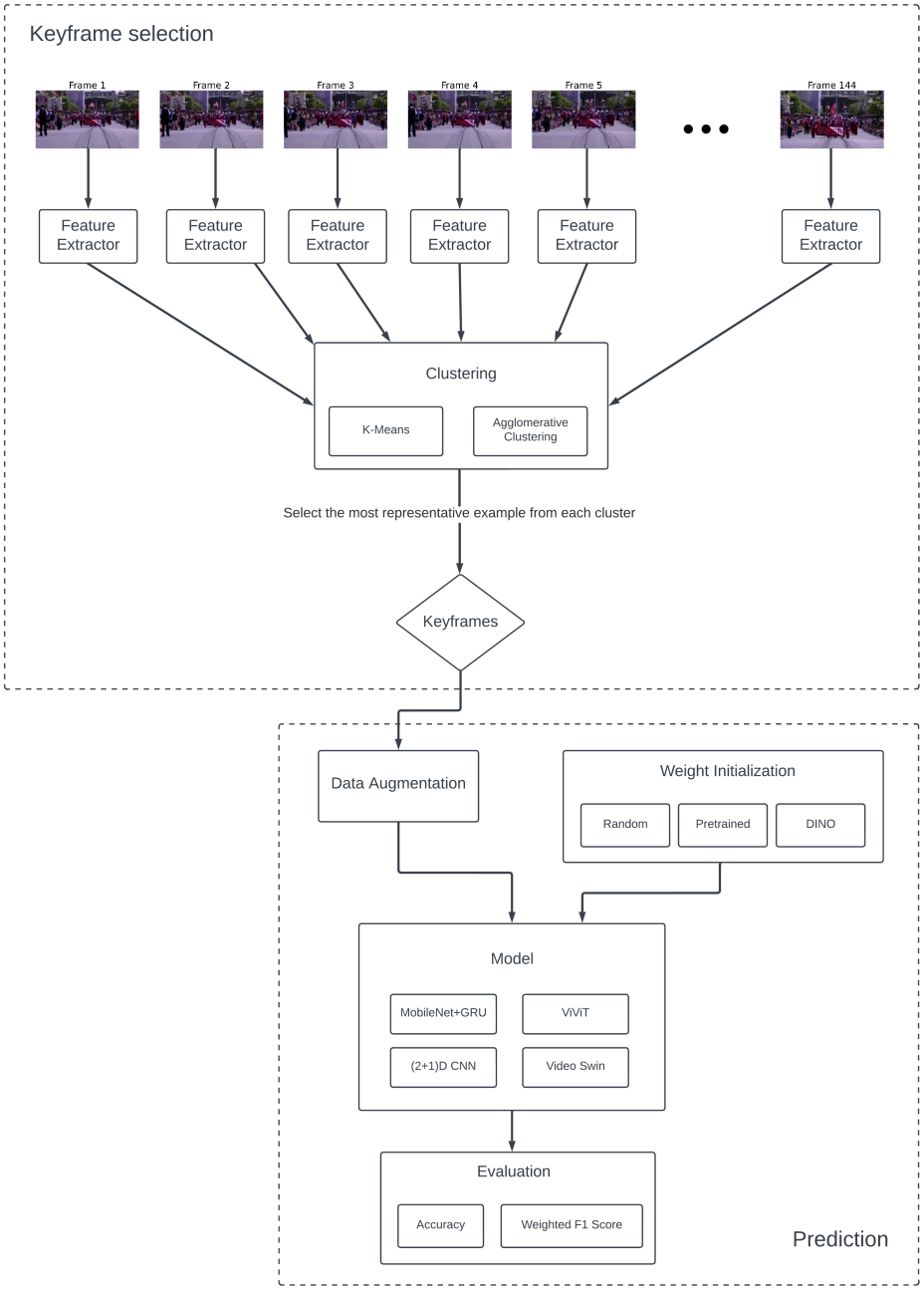

Figure 1: The proposed final pipeline. The pipeline first uses frame-level feature extraction, further performing clustering on the extracted features and selecting the keyframes. The keyframes are then used by the model to classify the sample.

The data augmentation plays a massive role as a regularization technique. It is increasingly important to have an effective regularization when

working with videos, because the video dataset are always marginally smaller in size than textual or image datasets meaning that data is scarce.

Usage of pretrained weights of existing image models has shown to have massive impact on models performance while decreasing the needed training time (Arnab et al., 2021; Liu, Ning, et al., 2021; Simonyan & Zisserman, 2014). Given the limited amount of data, pretrained weights help leverage the knowledge that is contained in large-scale image or video models.

A number of neural network architectures are used and compared, including baseline models of GRU+MobileNet and (2+1)D CNN, with Video Swin transformer showing to be the most preferential in terms of complexity and performance.

The models are being evaluated using the accuracy and weighted F1 score metrics.

3.2 *Dataset description*

The dataset used in this research is UCF-101 (Soomro et al., 2012). The dataset consists of over 13 000 clips, which is 27 hours of video data. There are 101 action classes/labels, which can be divided into following groups: Human-Object Interaction, Body-Motion Only, Human-Human Interaction, Playing Musical Instruments, Sports. The dataset has large variations in the movement of the camera, objects scale or camera position. The videos are sampled from YouTube, at the 320×240 pixels resolution and 25 frames per second (FPS). Videos are of different length. Authors provide three train-test splits, each consisting of around 9500 train samples and 3500 test samples. However, only the first train-test split was used because training and evaluating on two additional train-test splits heavily increases the required computational power. The dataset does not contain any irregularities or missing values.

3.3 *Keyframe selection*

Videos are sequences of frames, most of the time captured at rather short periods of time (measured in frames per second, FPS, with usual values for TV or YouTube videos in the range of 25 to 30 FPS). Therefore, it would be rational to assume that a lot of information contained in videos is redundant, as usually there is just a subtle difference between frames, and the high FPS in videos is required to achieve the effect of seamless motion, while information in the video can be summarised using a much smaller number of frames.

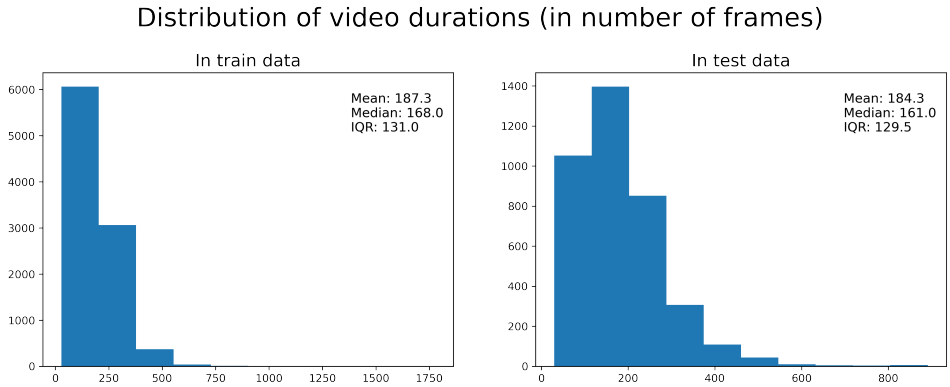

Figure 2: The distribution of durations of videos in the train and test set of UCF-101.

Keyframe selection is even more important when modelling large videos, because long videos makes it impossible to model the video completely and the videos have to be cropped, divided into parts and processed separately, or only a set of frames should be used.

Another application of keyframe selection emerges when dealing with videos of varying length. As usually machine learning algorithms do not accept data of inconsistent format, it is important to have the data aligned.

One solution is to trim and pad the videos to the same length, but this can be limiting in some scenarios. When dealing with long videos, it will most likely be the case that the videos will be heavily trimmed, leading to information loss. In cases when dealing with datasets that include videos with a wide range of durations, many videos would end up having a lot of redundant datapoints due to padding (e.g. having many zero-valued frames if zeros are used as padding). Although carrying little information, these datapoints would still take computational resources like RAM and CPU usage. Figure 2 shows that the distribution of video durations in both train and test set of UCF-101 is closer to exponential, with interquartile ranges of 131.0 and 129.5 respectively. This means that the durations of videos in the dataset varies largely and it is hard to manually pick a meaningful value of frames to use in the model.

However, keyframe selection offers the best of both worlds, when it is possible to both preserve the most crucial information while keeping the data size relatively small.

Gawande, Hajari, and Golhar (2020) explore a large number of modern-day keyframe selection methods. They highlight three categories of keyframe selection methods based on the keyframe size estimation. "Priori knowledge base as a fixed number" methods estimate the number of keyframes using a priori knowledge, while "Posteriori knowledge base

as unknown" algorithms allow for a dynamic keyframe size. As for this project it is crucial to utilize the knowledge of pretrained models that have been trained on a specific number of frames, it is important to satify following criteria:

1. Keep the number of frames constant
2. Be able to define the needed number of keyframes

Thus, only the "Determined-fixed number" keyframe selection algorithms can be considered.

In the overview, Gawande et al. (2020) highlight Cluster-based keyframe selection as the oldest method, with the first work appearing as early as 1998 (Zhuang, Rui, Huang, & Mehrotra, 1998). The authors note that, if performed on short videos, one of the methods advantages is the fast speed and the possibility to capture the global characteristics of the video. Although it is important to notice that the method does not capture the temporal dependency, as it only takes into account the frames' individual similarities and differences, not accounting for any temporal interaction per se. It is also possible to choose from a wide range of clustering algorithms, giving the possibility of a trade-off between performance and computational complexity.

Additionally, the method is flexible because before clustering the frames, some method of feature extraction is needed to create a representation of the frames that can be well utilized by the clustering algorithm. It can be both an established traditional algorithm from computer vision like color histograms or feature extraction using a machine learning model. For example, Gowda, Rohrbach, and Sevilla-Lara (2021) use a lightweight MobileNet (Howard, Zhmoginov, Chen, Sandler, & Zhu, 2018) as a feature extractor of the frames.

3.3.1 *Feature extraction*

As was mentioned earlier, it is impossible to directly feed the clustering algorithm with videos. As a preliminary step, feature extraction is needed. Feature extraction is the process of transforming the data into the format that is suitable for the future algorithm. Feature extraction should retain the information in the original data, while at the same time increasing the predictive power of the model.

The traditional methods of feature extraction in computer vision are well explored, so it might be more important to focus on using modern deep learning models that have been trained on a down-stream tasks. It is principal that the feature extraction phase is still not computationally

heavy, as the predicting step will already use a transformer model, which has a $O(n^2)$ time complexity.

MobileNet is a CNN-based model that was created to be as light as possible, while still having a close to SOTA performance. Although it uses convolutions instead of attention, it is important to include it in comparison, as MobileNet is a popular choice for feature extraction (Gowda et al., 2021). Similarly, it is important to include an image version of either ViViT or Video Swin, namely ViT (Dosovitskiy et al., 2021) and Swin (Liu, Lin, et al., 2021) as this way it is possible to compare whether using a similar architecture in both feature extraction and prediction phases influences the performance.

Feature extraction using a Neural Network is performed by taking the last layer that comes before the model's head. Usually, this layer will be the pooling layer.

By doing feature extraction on all of the frames, the resulting video of size

$$(N_{Frames}, HEIGHT, WIDTH, CHANNELS)$$

is transformed into matrix of

$$(N_{Frames}, N_{Features})$$

where $N_{FEATURES}$ depends on the dimension size of the last layer of the respective model used for extraction (usually the number is 512, 768 or 1024).

3.3.2 *Clustering*

Clustering is the task of grouping data points together by using the features' of the data points and combining together the most similar data points. There exists a number of types of clustering methods, namely:

1. Partition-based/Centroid-based Methods
2. Hierarchy-based Methods
3. Distribution-based Methods
4. Density-based Methods

The approach to how the clusters are being computed differs heavily between categories. Some use the underlying distribution of the data, some utilize the areas of the density. Furthermore, because the methods tackle the clustering problem differently and using different information, it is important to choose the suitable clustering algorithm. In this specific case,

it is rather hard to go with the Distribution-based clustering algorithms, as those require the knowledge of the distribution of the data (which is rather obscure in our method). The videos will be transformed into features using the neural networks, leaving us with large multi-dimensional matrices, while density-based algorithms have problems with high dimensional data. It is also important to use clustering algorithms that permit selecting the number of clusters, as we are required to be able to have some constant number of keyframes from every video.

K-MEANS The K-means clustering algorithm is a type of Partition-based algorithms. It aims to partition the space of observations into K clusters. The points are assigned to a cluster based on their proximity to the clusters' centroid. K is hyperparameter that controls the number of clusters generated by K-means and gives possibility to select a specific number of keyframes to extract.

AGGLOMERATIVE CLUSTERING Hierarchical Agglomerative Clustering is a type of clustering that builds a cluster hierarchy. Initially, all the data samples are of their own cluster and the clusters are being joined together in each new level of the hierarchy. The joining is based on the metric that is used to calculate the similarity of samples.

CHOOSING THE MOST REPRESENTATIVE EXAMPLE When the frames are grouped into clusters, it is required to select a single data sample from each cluster to act as the most representative example. The most representative examples would then denote keyframes. It is important to define a metric to select the most representative sample. The defined metric should be algorithm-agnostic, since different clustering methods are used and compared. For example, it could be possible to use the distance to the cluster's centroid as a metric, but it is only available for K-Means.

However, it is possible to define a metric that is sound using the cosine similarity measure.

$$\text{cosine similarity}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$$

The sample that is the most representative of the cluster is the one which has the maximal sum of cosine similarities between itself and other samples in cluster.

$$\underset{\mathbf{d} \in C}{f(\mathbf{d})} = \sum_{\substack{\mathbf{c} \in C \\ \mathbf{c} \neq \mathbf{d}}} \frac{\mathbf{c} \cdot \mathbf{d}}{\|\mathbf{c}\|\|\mathbf{d}\|}$$

3.4 *Data Augmentation*

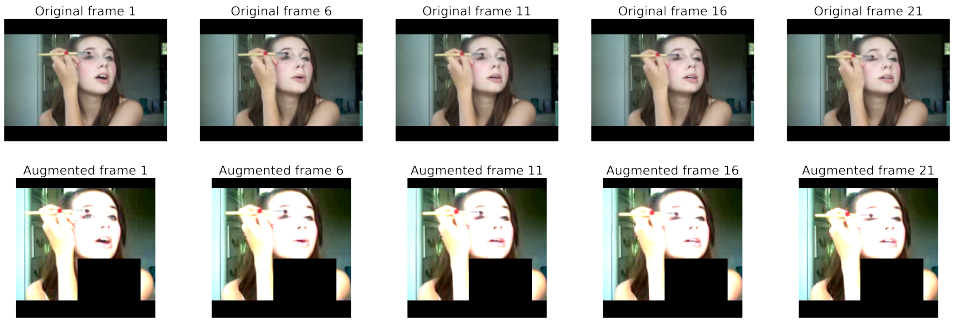

Figure 3: Example of the original and augmented frames of a video. On the augmented frames, the random jitter can be noticed, as well as the random erasing. The black bars on the edges of the frames represent the zero-padding to ensure consistent widths and heights of the videos.

Data augmentation acts as an additional regularizing constraint and helps extract additional data from the already existing one. For each of the models, the data augmentation pipeline consists of random cropping, color jitter, random horizontal flip and erasing of random small spatial regions of the video. These are regular data augmentation transformations that are widely applied (Arnab et al., 2021; Liu, Ning, et al., 2021; Simonyan & Zisserman, 2014). Figure 3 shows an example of the data augmentation transformations applied to a single video. The data augmentations are implemented to be temporally consistent, meaning that the same transformation is applied for all frames of the video. Some transforms as vertical flip are not used, as these would break the semantic information contained in the videos.

In addition to the mentioned data augmentation transformations, videos are normalized before being used as this improves model's converging time. For the models with pretrained weights, videos are normalized using the mean and standard deviation values of the original model.

3.5 *Label Smoothing*

Originally proposed by Szegedy, Vanhoucke, Ioffe, Shlens, and Wojna (2016), label smoothing is a regularization technique that regularizes the prediction of the classification layer and combats overconfidence. Label smoothing mixes uniform distribution with the original one-hot encoded labels. Original one-hot encoded labels are therefore swapped out using:

$$y_{LS} = (1 - \lambda) \times y_{OHC} + \frac{\lambda}{K}$$

where K is the number of classes in the dataset and $\lambda \in [0, 1]$ is a scalar hyperparameter.

3.6 *Models*

Before further exploration, it is first important to settle down with a concrete baseline model and a number of transformer-based models for prediction, and further it is possible to couple it with additional processing steps.

Initially, two baseline models are proposed. Additionally to the baseline models, two transformer-based architectures are explored.

3.6.1 *Baseline model*

Transformers are rather new and promising technology, that has only recently gained traction in academia and business. So to see and understand the true performance of the transformer models, it is important to establish a baseline that comes from a more traditional existing architecture that has already proved results.

EfficientNet family of models is based on the lightweight MobileNet v2 (Howard et al., 2018) model and have shown to achieve SOTA performance among CNNs for ImageNet benchmark, while being 5-10x faster and smaller than their rivals (Tan & Le, 2019). EfficientNet authors also emphasise the models' significant ability for transfer learning.

However, throughout this study, the EfficientNet+GRU model turned out to be computationally very expensive. In spite of computational effectiveness of GRU in comparison to LSTMs and EfficientNet's transcendence when compared to other popular CNN architectures, it still fails to be a feasible model as the training time is rather long. Therefore, the MobileNet v3 Large (Howard et al., 2019) is chosen, which has x10 less parameters than EfficientNet while still being a well-round model.

Based on this information, it is viable to say that the MobileNet+GRU and the (2+1)D models are fairly well-performing popular options as the current solutions to action recognition problem and thus can serve well as baselines.

3.6.2 *Video Vision Transformer (ViViT)*

ViViT was the first attempt to train a large image transformer model and is considered as the starting point of transformers in image domain. As was mentioned earlier, four methods of modelling temporal attention were proposed.

Attention is the basic and most important building block of transformers, so the difference in its computation heavily impacts performance and computational complexity. Authors find that factorised encoder approach gives the best performance on smaller datasets and is one of the best in terms of runtime and floating points operations (FLOPs). However, on large-scale datasets like Kinetics 400, spatio-temporal attention has better accuracy.

3.6.3 *Video Swin*

Video Swin model is largely influenced by ViViT, but overcomes some of ViViT's limitations. Due to the use of shifted windows method, it is able to compute only local attention which heavily decreases computational complexity and helps better model both temporal and spatial dependencies. Subsequently, it overcomes ViViT in both speed and accuracy.

However, contrary to ViViT, Video Swin creators conclude that the joint spatio-temporal attention models perform better. Based on the conclusions made in the works of Arnab et al. (2021) and Liu, Ning, et al. (2021) and limited timeframe of this project, only the joint spatio-temporal attention models are used.

Video Swin was trained on videos of 32 frames, so because of this, all the models (except for one) were trained on 32 frames.

3.7 *Weights Initialization*

In almost every setup pretrained weights are being utilized. This is needed because training a deep neural network requires substantial amounts of data. If coupled together with the fact that videos are themselves contain enormous amount of information and the complexity of the transformer architecture, then we will find ourselves requiring the need of large computational clusters to train the model.

For most of the models like (2+1)D CNN, MobileNet and Video Swin, the default weights provided by respective model authors are used.

For ViViT, the weights of image transformer DINO (Caron et al., 2021) are used. DINO is a self-supervised approach for training Vision Transformers which outperforms original ViT on ImageNet benchmark. The embedding layer of the DINO consisting of a 2D CNN is inflated using central-frame initialization method introduced by Arnab et al. (2021).

The MobileNet V3 Large, (2+1)D CNN and DINO models were pretrained on the ImageNet-1K dataset.

The Video Swin model was pretrained on Something-Something V2 video dataset.

3.8 *Evaluation Metrics*

As for most classification tasks, Top-1 accuracy is the most common metric for video action recognition because of its easy interpretability.

F1 score is another metric that is regularly used for evaluating classification models.

3.9 *Implementation Details*

All aforementioned data loading, preprocessing, keyframe selection, data augmentation and model creation was performed using Python and libraries numpy, scipy, scikit-learn, cuML, torch, torchvision and transformers. Graphs were generated using matplotlib and seaborn libraries.

4 RESULTS

The results section aims to explore and compare the performances of the baseline models, the performance of the ViViT and Video Swin models. Additionally to comparing the bare models, the collation of different clustering algorithms and feature extractors is presented. Not only is the performance of the setups is looked at, but also their computational effectiveness. At the end of the results section, a a more thorough analysis of the final setup is further explored, with a breakdown of the predictions and prominent shortcomings.

4.1 *Baseline*

Model	Weights Init.	Accuracy	W-F1
GRU + MobileNet V3 Large	Mixed	57	0.56
(2+1)D CNN	Pretrained	77	0.80

Table 1: Table with the results of the baseline model. The reported metrics are for the test set. GRU+MobileNet V3 Large was initialized with pretrained weights for MobileNet and random for GRU.

Table 1 compares the performance of the baseline models. Looking at the accuracy and F1 scores, it can be clearly seen that (2+1)D CNN setup outperforms GRU+MobileNet. Most likely, this can be attributed to the fact that the recurrent GRU model was trained from the beginning and only the weights for MobileNet were pretrained, while the (2+1)D initializes all weights from a pretrained model. Due to the superiority of (2+1)D CNN

baseline model compared to the GRU+MobileNet model, only the (2+1)D is further used for comparison.

4.2 *ViViT*

Model	Weights Init.	Frames	Accuracy	W-F1
ViViT	Random	32	31	0.28
ViViT	DINO	32	76	0.75
ViViT	DINO	32 keyframes	85	0.85
ViViT	DINO	10 keyframes	83	0.83

Table 2: Table with the results of the models with ViViT backbone. The reported metrics are for the test set. The keyframes were extracted using MobileNet features and K-Means clustering.

ViViT was initialized either using random or pretrained DINO (Caron et al., 2021) weights. Based on the Table 2, it can be well seen that the initial setup with random initialization of the weights performed far worse than the baseline, most likely due to the minuscule amount of training data, compared to the usual needs of transformer models. In case of pretrained DINO weights, the 2D embeddings of DINO were inflated into third dimension, meaning that the embedding weights were not optimally pretrained at the start. Although getting very close, ViViT with DINO weights still did not manage to overcome the baseline (2+1)D CNN. However, the use of keyframe selection algorithm has greatly increased the predictive power of the ViViT model, achieving substantial 85% accuracy.

An interesting finding is that using just 10 keyframes lead to a large increase in accuracy compared to the 32 frame model without keyframe selection, achieving 83% accuracy while using just a third of the previous amount of frames. This is an important finding not only because of the increased accuracy, but also because of the reduced computational load.

4.3 *Video Swin*

Video Swin model performed the best both as a bare model and with keyframe selection. Table 3 shows that by utilizing 32 keyframes selected using Agglomerative clustering, it is possible to achieve the highest performance across all setups. Partly, it can be attributed to the fact that Video Swin is the only model that was initialized with completely pretrained weights. However, additionally to the incredible performance of the model, Video Swin computes only local attention instead of global attention that is used in ViViT. The use of local attention, means that Video Swin has not

Model	Weights Init.	Frames	Accuracy	W-F1
Video Swin	Pretrained	32	93	0.93
Video Swin	Pretrained	32 keyframes (K-Means)	94	0.94
Video Swin	Pretrained	32 keyframes (Agg.)	95	0.95

Table 3: Table with the results of the models with Video Swin backbone. The reported metrics are for the test set. The keyframes were extracted using MobileNet features and the mentioned clustering algorithm.

only the highest accuracy, but also requires less computational resources than other transformer-based setups.

4.4 *Feature extractors*

Extractor	Frames	# parameters	$N_{Features}$	Accuracy
Swin	32 keyframes	5.5M	960	94.7
MobileNet V3 Large	32 keyframes	27.5M	768	95.4

Table 4: Table with comparison of using different feature extractors for further clustering. The reported metrics are for the test set using the best setup with Video Swin model (pretrained weights initialization) and agglomerative clustering. $N_{Features}$ represents the number of features that the feature extractor outputs per each frame of the video.

Table 4 shows comparison of using different frame-level feature extractors that are used to generate features for further clustering. It can be seen that MobileNet slightly outperforms Swin features, but that can be also attributed to chance. MobileNet produces a larger number of features, which can have result in a slightly slower performance when performing clustering, but can also lead to better features due to the additional information.

After reviewing the available information, it can be concluded that both feature extractors generate well-performing features that produce keyframes that can be well utilized by a Machine Learning model. However, the Swin model contains 27.5 millions of parameters, while MobileNet V3 Large has just 5.5 millions, being five times smaller than its transformer counterpart. This large difference in the number of parameters means that the feature extraction step will take more time and computational resources. This leads to choose MobileNet as the feature extractor because of its superiority in size and therefore also in speed.

4.5 *Analysis of Prediction Errors*

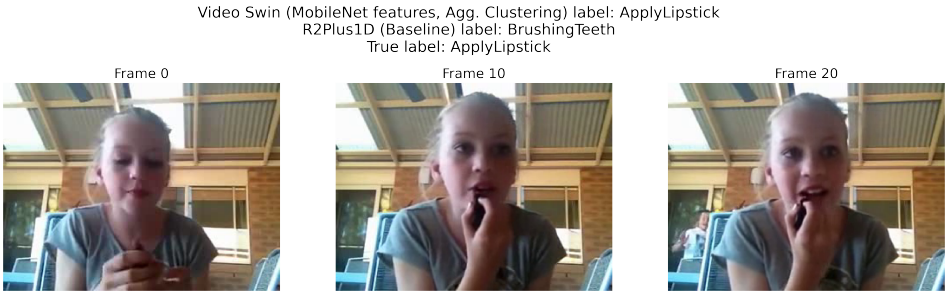

Figure 4: An example where the Video Swin model classified the sample correctly, while the baseline (2+1)D CNN failed.

When comparing the final best-performing transformer model with the baseline (2+1)D convolutional model, it is both interesting and important to not just compare bare score, but have a more in-depth analysis of the differences of their predictions. Figure 4 shows how the Video Swin model managed to correctly classify a video of a girl applying lipstick, while the baseline model seen it as teeth brushing. The actions are indeed similar in their visual appearance, however the Video Swin model might have based its prediction on, for example, the angle at which the girl holds her hand (which in case of teeth brushing would be more horizontally rotated).

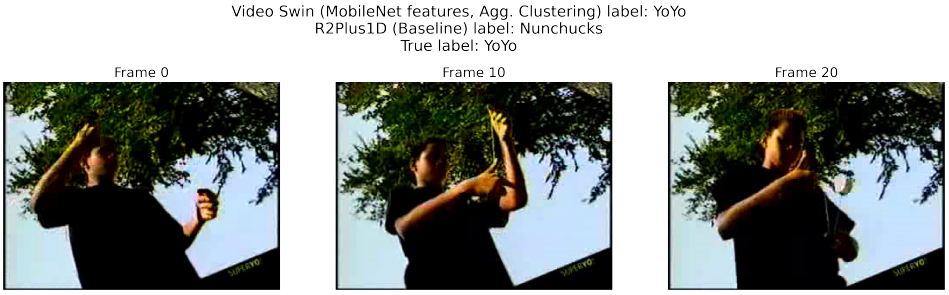

Figure 5: Another example where Video Swin outperforms the baseline.

Figure 5 shows another sample, where, when not paying attention to the details, some confusion may arise. The Video Swin flawlessly predicted the correct "Yo Yo" action, while the (2+1)D CNN has seen the use of nunchucks. When looking at the frames, one might find similarity between the boy's hand position and movement and the usual movement of hands operating nunchucks. However, the cue to the correct answer is again in the details which the baseline did not recognize: a Yo Yo's string and the way it is operated might indeed have some similarity to the nunchuks,

but the Yo Yo's string that is visible on the video is much thinner than nunchucks.

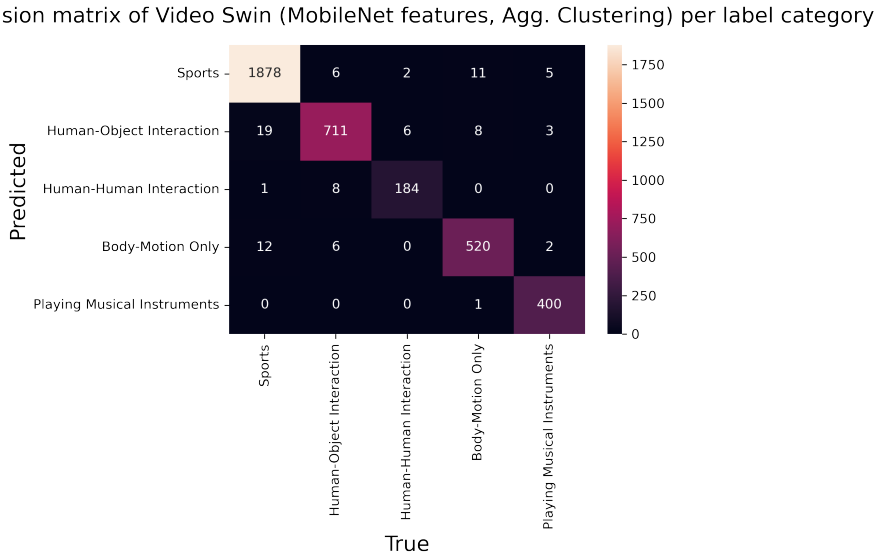

Figure 6: Confusion matrix of the Video Swin predictions with labels combined into categories.

As was mentioned initially in the dataset description, the labels in the UCF-101 can be combined into five groups (provided by the dataset authors). From the confusion matrix on Figure 6, it can be seen that the most troublesome part for the Video Swin model comes from making a difference between the "Sports" activities with "Human-Object interaction" and "Body-Motion only" activities. The overall visual appearance of the activities from those groups can indeed be sometimes hard to spot, as the activities that we have in "Sports" can also be well described by phrases as "Human-Object interaction" and "Body-Motion Only". This might mean that the model does not always pick up the semantical information of the individual videos, leading to classification. However, this can also be explained by the fact that the dataset is largely imbalanced, with the bigger part of data samples being of "Sports" label.

Table 5 highlights some of the most frequent errors of the Video Swin model.

For example, when looking at the labels that Video Swin model mispredicted for the "CricketShot" label, it turns out that instead of the correct "CricketShot" label the model predicted 16 times the label "CricketBowling". As the name suggests, both of the actions involve the sport of cricket, so the model has a hard time distinguishing between different actions in the same sport.

True label	Predicted	Amount
CricketShot	CricketBowling	16
	HammerThrow	2
Shotput	CricketShot	6
	ThrowDiscus	3
	HammerThrow	3
	CricketBowling	2
	JavelinThrow	1
Nunchucks	GolfSwing	10
	SalsaSpin	1
	YoYo	1
Hammering	BabyCrawling	3
	MoppingFloor	2
	BlowDryHair	2
	BrushingTeeth	1
	ShavingBeard	1
	Haircut	1
	HandstandWalking	1
JumpRope	GolfSwing	2
	Nunchucks	2
	TaiChi	2
	SoccerJuggling	2
	Lunges	1
	TennisSwing	1

Table 5: The comparison of the top-5 mispredicted labels and the labels which were predicted instead.

For the label "Shotput", the most of the errors come from predicting "CricketShot", "ThrowDiscus" and "HammerThrow". The sports of shot put, disc throw and hammer throw are all members of the throwing sports. The "CricketShot" label refers to the act of hitting the ball in cricket, meaning the dynamics of the body movement is very similar to that of shot put.

All combined, it can be concluded that although the Video Swin setup performs reasonably well, it is still not always able to pickup the small details in the videos, leading to misclassifications.

A rather strange finding, is that three times, a video with "Hammering" label was predicted as "BabyCrawling" although it is hard to find similarities between those activities.

5 DISCUSSION

The final goal of this study was to find a machine learning pipeline for action recognition that includes a transformer-based machine learning model and would not require large datasets or immense computational power, while still retaining the ability to be an effective predictor.

The goal was addressed successfully by creating a pipeline that includes a Video Swin transformer-based model as the backbone, while still focusing on performance by utilizing pretrained models and effective keyframe selection.

One of the baseline models has set a strong baseline of 77% accuracy. Overcoming it was not possible even with the first two transformer-based pipelines. The poor performance of the pure ViViT setup can be mostly attributed to the problem that ViViT authors highlight themselves - the architecture is the first trial of using transformers for videos and thus the technology did not yet mature enough to be optimized (Arnab et al., 2021). However, using keyframe selection methods managed to greatly improve the performance of the base ViViT model. A particularly interesting finding is that the 10-keyframe ViViT setup has largely outperformed the baseline models and even the 32-frame ViViT setup (without keyframe selection algorithm). Partly, it can be explained by the possible noise present in the videos. Keyframe selection with such low amount of extracted keyframes is able to retrieve only the most important and descriptive frames, making any noise negligible. This, when compared to the 32-keyframe ViViT, gives the possibility to have a trade-off between accuracy and speed. 10-keyframe ViViT model has only 2% decrease in accuracy compared to the 32-keyframe ViViT, however the training time decreased from eight hours (for both plain ViViT and 32-keyframe version) to just one hour.

The significantly impactful performance boost was achieved by using the Video Swin model. As Video Swin is the next generation of video transformer models, it overcomes some of the shortcomings of the initial ViViT. Paired with Agglomerative Clustering, the setup achieved 95% accuracy on the test set. Transition from K-Means clustering has improved the accuracy by 1%, which, although a speculative conclusion due to the minor increase in performance, might mean that the space of video frames is better representable using hierarchies rather than partitions that are used in K-Means clustering. Another important remark about the training of the Video Swin is that it took substantially less amount of epochs to train the model compared to ViViT. The training loss for Video Swin has saturated already after 5-7 epochs, while ViViT's training loss continued to improve slightly even after 20 epochs. This might signal that the pretrained weights initialization of the Video Swin weights is more optimal than that of DINO

weights used to initialize ViViT. However, this can be explained by the fact that Video Swin weights were completely pretrained on a large video dataset contrary to the weights used for ViViT.

Overall, although the final accuracy of 95% is close to the perfect 100%, there is still room for improvement. Similarly to the self-supervised approach of DINO image transformer training (Caron et al., 2021), it would be interesting to see how will the performance of Video Swin model increase if it is first pretrained in self-supervised or semi-supervised manner. As was stated earlier, labelled datasets in video domain are particularly valuable due to their insignificant amount, so finding a method of self-supervised video transformer training is an important milestone.

Coming back to the initially set research questions, it can be concluded that all of them were addressed and backed by evidence.

The main research question *"To what extent can transformer-based models be used for activity recognition on small datasets?"* was acknowledged by successfully training a well-performing Video Swin model that overcomes the baseline, while having a decreased computational complexity due to the use of local attention. Considering that the limited size of the dataset is a large constraint, it was important to find additional sources of information, which was done by using such regularization techniques as label smoothing and data augmentation. Additionally to regularization, the usage of pretrained weights has greatly helped reduce the needed training time and improve accuracy.

The research subquestion *"To what extent can image transformers be used to improve classification of videos"* was answered by showing significant improvement that the ViViT model with DINO weights has over the ViViT model with random weight initialization. Going further, it is possible to conclude that successful training of such large video transformer models as ViViT using datasets of 10,000 videos such as UCF-101 is impossible. Simply put, the transformer architecture without any additional pretraining is not the optimal solution when the data is limited.

The great impact of keyframe selection can be seen when comparing the results for the ViViT model. If keeping the number of frames that is fed into the model constant, using keyframe selection yields additional 10% accuracy. However, keyframe selection is also beneficial because of its possibility to effectively summarize the video. This opens the door to reducing the size of the data by using less frames, instead trading it for a smaller portion of accuracy. Therefore, it can be concluded that the research subquestion *"To what extent selection of key frames affects performance of video classification"* has also been addressed.

6 CONCLUSION

In this study, the methods of human action recognition on videos were explored and compared. The final setup, consisting of Video Swin transformer model combined with MobileNet feature extractor and keyframe extraction using Agglomerative clustering, has shown to be the best performing one and largely overcoming the baseline. The study has managed to successfully address the research questions that were set at the beginning, and provide solutions and evidence of their performance.

Although the overreaching goal of the study was mainly framed around usage of transformer-based models, the study has also managed to show the importance of frame sampling when working with videos. Possibly, this can lead to better utilization of the computing power in future research, by, for example, embedding the frame sampling stage as part of the transformer model, possibly by utilizing the same self-attention mechanism.

Finding a self-supervised or semi-supervised training approaches for video transformer models can be seen as possible prospective direction for future studies.

A CODE

The relevant programming code produced throughout this thesis is publicly available at `https://github.com/jegork/bachelor-thesis`

REFERENCES

Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., & Schmid, C. (2021). Vivit: A video vision transformer. In *2021 ieee/cvf international conference on computer vision (ICCV)* (p. 6816-6826). doi: 10.1109/ICCV48922.2021.00676

Caron, M., Touvron, H., Misra, I., J'egou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 9630-9640.

Cho, K., van Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder–decoder approaches. In *SSST@EMNLP*.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Nips 2014 workshop on deep learning, december 2014*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 4171–4186). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://aclweb.org/anthology/N19-1423 doi: 10.18653/v1/N19-1423

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International conference on learning representations*. Retrieved from https://openreview.net/forum?id=YicbFdNTTy

Gawande, U., Hajari, K., & Golhar, Y. (2020). Deep learning approach to key frame detection in human action videos. In A. Sadollah & T. S. Sinha (Eds.), *Recent trends in computational intelligence* (chap. 7). Rijeka: IntechOpen. Retrieved from https://doi.org/10.5772/intechopen.91188 doi: 10.5772/intechopen.91188

Girdhar, R., Singh, M., Ravi, N., van der Maaten, L., Joulin, A., & Misra, I. (2022). Omnivore: A single model for many visual modalities. *CoRR*, *abs/2201.08377*. Retrieved from https://arxiv.org/abs/2201.08377

Gowda, S. N., Rohrbach, M., & Sevilla-Lara, L. (2021, May). Smart frame selection for action recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(2), 1451-1459. Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/16235

Hochreiter, S., & Schmidhuber, J. (1997, 12). Long short-term memory. *Neural computation*, *9*, 1735-80. doi: 10.1162/neco.1997.9.8.1735

Howard, A., Sandler, M., Chu, G., Chen, L., Chen, B., Tan, M., ... Adam, H. (2019). Searching for mobilenetv3. *CoRR*, *abs/1905.02244*. Retrieved from http://arxiv.org/abs/1905.02244

Howard, A., Zhmoginov, A., Chen, L.-C., Sandler, M., & Zhu, M. (2018). Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. In *CVPR*.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *CVPR*.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., ... Zisserman, A. (2017). The kinetics human action video dataset. *CoRR*, *abs/1705.06950*. Retrieved from http://arxiv.org/abs/1705.06950

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., & Serre, T. (2011). HMDB: a large video database for human motion recognition. In *Proceedings of the international conference on computer vision (ICCV)*.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*(4), 541-551. doi: 10.1162/neco.1989.1.4.541

Li, K., Wang, Y., Peng, G., Song, G., Liu, Y., Li, H., & Qiao, Y. (2022). Uniformer: Unified transformer for efficient spatial-temporal representation learning. In *International conference on learning representations.* Retrieved from https://openreview.net/forum?id=nBU_u6DLvoK

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 9992-10002.

Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., & Hu, H. (2021). Video swin transformer.

Mordor Intelligence. (2021). *Natural language processing (nlp) market - growth, trends, covid-19 impact, and forecasts (2022 - 2027).* Retrieved from https://www.mordorintelligence.com/industry-reports/natural-language-processing-market

Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Proceedings of the 27th international conference on neural information processing systems - volume 1* (p. 568–576). Cambridge, MA, USA: MIT Press.

Soomro, K., Zamir, A. R., & Shah, M. (2012). Ucf101: A dataset of 101 human actions classes from videos in the wild.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (p. 2818-2826). doi: 10.1109/CVPR.2016.308

Tan, M., & Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR*, *abs/1905.11946*. Retrieved from http://arxiv.org/abs/1905.11946

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jegou, H. (2021, 18–24 Jul). Training data-efficient image transformers amp; distillation through attention. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th international conference on machine learning* (Vol. 139, pp. 10347–10357). PMLR. Retrieved from https://proceedings.mlr.press/v139/touvron21a.html

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., & Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In *2018 ieee/cvf conference on computer vision and pattern recognition* (p. 6450-6459). doi: 10.1109/CVPR.2018.00675

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L.,

Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In I. Guyon et al. (Eds.), *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Wang, R., Chen, D., Wu, Z., Chen, Y., Dai, X., Liu, M., ... Yuan, L. (2021). BEVT: BERT pretraining of video transformers. *CoRR*, *abs/2112.01529*. Retrieved from https://arxiv.org/abs/2112.01529

Zha, S., Luisier, F., Andrews, W., Srivastava, N., & Salakhutdinov, R. (2015). *Exploiting image-trained cnn architectures for unconstrained video classification.* Swansea, UK.

Zhuang, Y., Rui, Y., Huang, T., & Mehrotra, S. (1998). Adaptive key frame extraction using unsupervised clustering. In *Proceedings 1998 international conference on image processing. icip98 (cat. no.98cb36269)* (Vol. 1, p. 866-870 vol.1). doi: 10.1109/ICIP.1998.723655