Comparative Analysis of ViViT for Video Action Recognition

Pranav Arora
Aalto University
02150 Espoo
pranav.arora@aalto.fi

Hans Tiwari
Aalto University
02150, Espoo
hans.tiwari@aalto.fi

April 21, 2023

Abstract

Transformers have gained significant popularity in NLP and with image classification (Vision Transformer). Video action recognition entails identifying the action in a video based on the input and labels. In this paper we aim to perform a comparative analysis on video vision transformers released by Google on large scale and small dataset with different training setting to evaluate their performance. Specifically we target Activity-net and UCF101 for conducting our experiments. The primary architecture of ViViT, factorised encoder is used to conduct our experiments. We evaluate the experimental results based on training the model without pretraining, adding some pre trained layers such as positional embedding and patch embedding from ViT initialized weights and using the pre-trained ViViT trained in large scale data such as kinetics. We evaluate our results by drawing out confusion matrices which helps in evaluating the predictive capability of models per class.

Keywords : Transformer, UCF101, Activity-net, ViViT, pre-training, action recognition

1 Introduction

Video action recognition [1] is a task that involves the identification and classification of human actions or activities from a video sequence. It consists of extracting key features from a video or clip and then uses a machine-learning algorithm to perform the recognition task. Different actions or activities could cover running, walking, jumping, waving, and more. The applications of this task are endless, covering surveillance, sports analysis, healthcare, and entertainment. Accurate and real-time video action recognition can enable machines to understand and interpret human actions, leading to advances in human-computer interaction and artificial intelligence.

In ViViT architecture, transformers are the core part, so we elaborate on it in detail here. In a typical recurrent neural networks (RNNs), the input frame or text is processed sequentially one at a time. This can lead to limitations in parallel processing and covering long-range dependencies. Self-attention, on the other hand, allows a neural network to give more importance to the different frames and texts in a sequence when processing it with context. It also allows the machine learning model to attend to all the input sequences parallelly, that is, simultaneously.

For each input that we get, in the form of words and pixels, the self-attention mechanism calculates three vectors, that are, query, key, and value. The scores of the attention output represent the significance or relevance of one word or pixel with respect to the others. These scores are helpful in the final feed-forward neural network to generate the final output, by calculating the weighted sums of the values.

The Transformer architecture uses multiple self-attention layers stacked together, allowing the model to capture complex dependencies and interactions among different elements in the sequence. The self-attention mechanism [8] in the Transformer has several advantages, such as capturing long-range dependencies, enabling parallel processing, and allowing the model to focus on relevant information in the input sequence. These properties make the Transformer well-suited for a wide range of tasks, including machine translation, language modeling, image captioning, and video action recognition, among others.

Deep CNNs have been serving well and surpassed the state-of-the-art results on video action recognition. Transformer architecture has practically revolutionized the image domain as well, beating the state-of-the-art results by CNNs. But, video vision transformers(ViViT) have received state of the art results only when they are trained on a large scale dataset such as Kinetics 400, Epic Kitchens and Something-Something v2 [1]. This is primarily because transformers are very data hungry and require large scale datasets to initialise their inductive biases. For our experimentation we are using Activity net as a large scale benchmark. Although it is a large scale benchmarks there could be many possible best-practices for training. With convolution network the best-practices for training are already pretty concrete, as researchers and developers have been working on deep CNNs for years now. We start our experimentation by implementation Factorised Encoder ViViT and train it without any pretrained weights and then introduce the pretrained weights to compare the effect of different pre-training strategies. We follow the same steps to initiate our experiments with UCF-101 as well.

Previous works such as space-time attention [2] focused on using a single transformer which performs well on small scale but doesn't work well with large datasets, and requires more FLOPs.To overcome this the ViViT [1] authors develop three model where they factorize spatial and temporal transformer, factorised self-attention and factorised dot product attention. These model although have more number of parameters but they require less number of FLOPs, which makes these model more suitable for training on large scale video datasets.

The Vision Transformer (ViT) [5] is a cutting-edge deep learning architecture widely used for image recognition tasks. Recently, ViT has also been applied to pretraining video data, where it has shown promising results. By leveraging large amounts of video data, ViT can learn rich spatio-temporal representations of videos, which can then be fine-tuned for downstream tasks such as action recognition and video captioning. Pretraining ViT [1, 9] on video data has the potential to overcome the drawbacks of traditional 3D convolutional neural networks (CNNs) on video-related tasks, making it an niche area of research for the computer vision community

2 Model Architecture

2.1 Video Clips Tokenization and Embedding

For embedding the input to pass through the model we employ the two techniques as described in the ViViT paper [1]. A simple method used in all the video classification task for tokenization is Uniform sampling [1, 2, 8]. In uniform sampling N number of frames are sampled from an input video. These sampled non-overlapping frames are then used to create a patch of defined patch size(height and width). The whole process is shown in the figure 1. So for example if we use 224*224 height and width of a frame, then each frame is added to the patch which is defined by passing a patch size parameter which is set to 16. Hence, we obtain 14 patches of size 16*16 while performing the patch embedding. After that each of the patch is stacked together to be passed to the positional embedding layer.

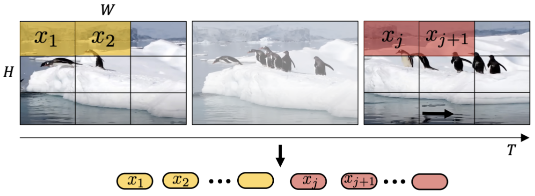

Figure 1: Uniform frame sampling to embed each frame in a video sample[1]

Another method we employ is the tubelet embedding. In figure 4 and figure 2, instead of extracting frames, spatio-temporal tubes from input is extracted and then linearly projected using a 3D-convolution [1]. Smaller tubelet size results in more number of tokens. In theory, this embedding method fuses the spatio-temporal information before the the input is passed to the model whereas in case of the uniform sampling method, the temporal information is fused to the input when it passes through from spatial to temporal transformer. Here converting the input video to tube a 3-D convolution layer is used. For example if we have a video of size 32*3*224*224, then using a stride and kernel of (2,16,16), we get a 3-D tube of size (16,14,14).

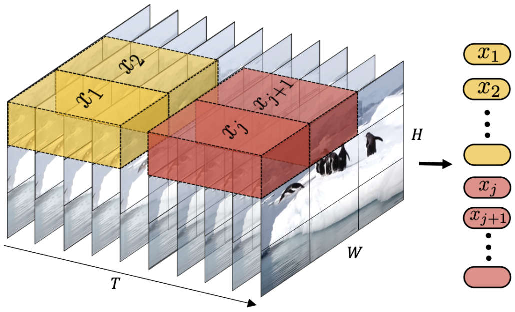

Figure 2: Tubelet embedding to embed spatio-temporal tube from a video sample [1]

2.2 ViViT Factorised Encoder

The model architecture used is ViViT factorised encoder.[1] The model is based on transformer architecture.[8]. The model comprises of two distinct encoders, a spatial and a temporal encoder. The spatial encoder is only responsible for learning the interactions between tokens at a single temporal index. An embedded representation at spatial level is outputted from N number of spatial encoders. This embedded representation is then concatenated and temporal token embedding is added after which the representation is passed through the N number of temporal encoder layers. Each video is passed to the model as frames in a sequence which is then embedded to tokens. On This representation, firstly either mean or average pooling is applied after which the channel dimension is down scaled to the number of classes in the dataset using a MLP with layer normalization to classify the action in the video.

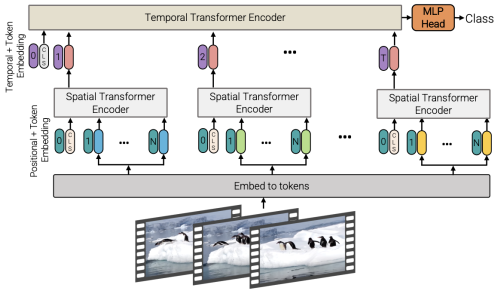
Temporal Transformer Encoder
MLP Head
Class
Temporal + Token Embedding
Spatial Transformer Encoder
Spatial Transformer Encoder
Spatial Transformer Encoder
Positional + Token Embedding
Embed to tokens

Figure 3: ViViT Factorised Encoder [1]

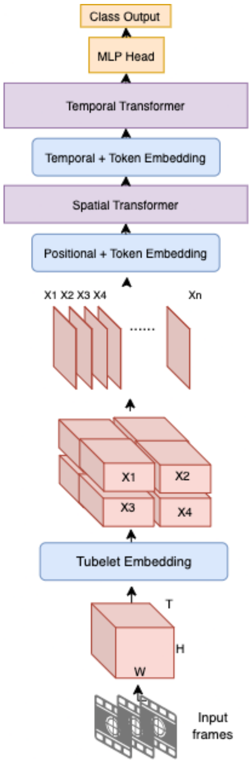

Figure 4: Tubelet Embedding Flow Diagram

3 Experiments

We conduct a series of experiments. This section describes the datasets and the training methodology.

3.1 Datasets

UCF101 containing 101 classes has been widely used as a benchmark dataset for action recognition research and has served as the basis for the evaluation of numerous state-of-the-art action recognition algorithms [7]. It has been used to advance the field of computer vision and machine learning, allowing researchers to develop and test new approaches for automatically recognizing human actions in videos. It is a small scale dataset for the video action recognition problem, especially in context of the transformers. It contains 9500 videos for training and 3000 videos in test set. Three separate split are provided by data-set authors to allow fair evaluation.

ActivityNet is a large-scale video understanding dataset that is designed to encourage the development of algorithms for human activity recognition in videos. It is a benchmark dataset that is widely used in computer vision research, particularly in the area of action recognition.

In this research, we use ActivityNet 200 [6] which contains 200 unique classes that are chosen to cover a broad range of human actions, such as ”brushing teeth”, ”riding horse”, ”eating sandwich”, ”playing guitar”, and many more. The dataset is divided into training, validation, and testing having around 10k, 5k, 5k videos in each. The number of samples used for the research is higher as on average there are three segments per video. We work with trimmed videos, hence the samples for training are more.

3.2 Training

For both of the dataset we use similar experiments to compare the performance of ViViT. As a baseline or first experiment we train the model from scratch with following hyper-parameters, batch-size : 32, learning-rate-0.01, optimizer:Adam, loss function - cross-entropy, crop-size: 224*224, patch size ; 16*16, epochs : 40. Secondly we use ViT [5] pre-trained weights trained on image net 30K for initialising the positional embedding. As a third experiment we use the ViT pre-trained weights for initialising the tubelet embedding weights and test the affect of adding pretrained weights at the positional embedding and tubelet embedding layer. As a final experiment we transfer the ViViT weights pre-trained on Kinetics 400 [1] for all the layers and fine-tune the model according to UCF1010 and activity-net. The ViViT weights for model trained on Kinetics 400 is converted from flax to PyTorch for this experiment.

The point of experimenting with different pre-training configuration is to observe the effect on the performance as we move from the baseline model to ViViT pretrained model. This helps us in mapping the accuracy progression along the whole training process and complementing on the fact whether some -layers initialised with pre-trained weights are enough to achieve accuracy comparable to the model which has pre-trained weights for each layer.

The final experiment of ViViT with pretrained weights is also conducted at different depth ranging from 4 to 12 to see the affect of increasing the number of spatial and temporal encoder in the the factorised attention model. In table 1 we can see that for UCF101, initially increasing the depth of the transformer really buttressed the accuracy but after reaching the depth of 10 it saturates and does not offer much improvement. For Activity-net the performance does keep on increasing with the depth. Such a difference can be attributed to the fact that the activity-net is a larger dataset as it has multiple clips under a single video file offering much more data and scope of improvement with depth when compared with the UCF101 dataset.

Depth	UCF101(%)	Activity-net(%)
4	43.59	48.27
6	56.47	59.34
8	72.31	64.18
10	94.73	73.94
12	94.35	85.54

Table 1: Affect on accuracy while increasing the depth of the model

4 Results and Conclusion

In Table 1, we can see the comparison of our model's variations for UCF-101 and ActivityNet 200. Adding

Model	**UCF101**	**Activity-net(Val set)**
MMViT[3]	95.99 (Three Fold)	-
SCT [9]	97.80 (Three Fold)	-
Vidtr [10]	96.4 (Three Fold)	-
scratch	7.58	11.02
pos-embedding(Vit Intialized)	25.53	16.54
pos embedding and tubelet embedding(Vit Initialized)	22.61	14.24
tubelet embedding(Vit Initialized)	16.54	13.83
ViViT pretrained weights w/o tubelet	37.84	22.16
ViViT pretrained weights with tubelet	**94.62**(Three Fold)	**85.54**

Table 2: Model Comparison on ActivityNet and UCF-101 Datasets

ViT pre-trained weights improved the performance by 18% in UCF and 6% in ActivityNet. The bigger difference is created by ViViT pre-trained weights. Initially, without tubelet weights the increase in performance was by 13% in UCF and 5% in ActivityNet from the previous best. After adding tubelet weights from ViViT the performance significantly improved by 57% in UCF and 63% in ActivityNet from the previous best. In figure 4 we can see the class-wise correct predictions made on test set of UCF101. It clearly shows that most of the classes are easily predicted but for some classes such as ”Shotput” , ” JumpRope” , the accuracy is not that good. The primary reason for such discrepancy is the quality of the videos, which can significantly affect the performance of the model.

To conclude, it is worth mentioning that tubelet representation which was generated using a 3-d convolution played a key role in improving the model accuracy as when the pre-training weights for the tubelet were added we observed a big difference in accuracy of the model.In conclusion, ViViT pre-trained weights work significantly better than ViT for the action recognition task.

5 Future Work

Further for more improvement, key frame selection can be used which could help in the reduction of noise and better performance of the current model. We have not played around with loss function while testing out variations of our model. In TCLR [4], contrastive loss is used with UCF-101 dataset, it shows promising results and if combined with ViViT, might help in improving performance.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021.

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021.

[3] Jiawei Chen and Chiu Man Ho. Mm-vit: Multimodal video transformer for compressed video action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1910–1921, 2022.

[4] Ishan Dave, Rohit Gupta, Mamshad Nayeem Rizve, and Mubarak Shah. Tclr: Temporal contrastive learning for video representation. *Computer Vision and Image Understanding*, 219:103406, 2022.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[6] Victor Ghanem Bernard Niebles Juan Carlos Heilbron, Fabian Escorcia. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR2015*, pages –, 2015.

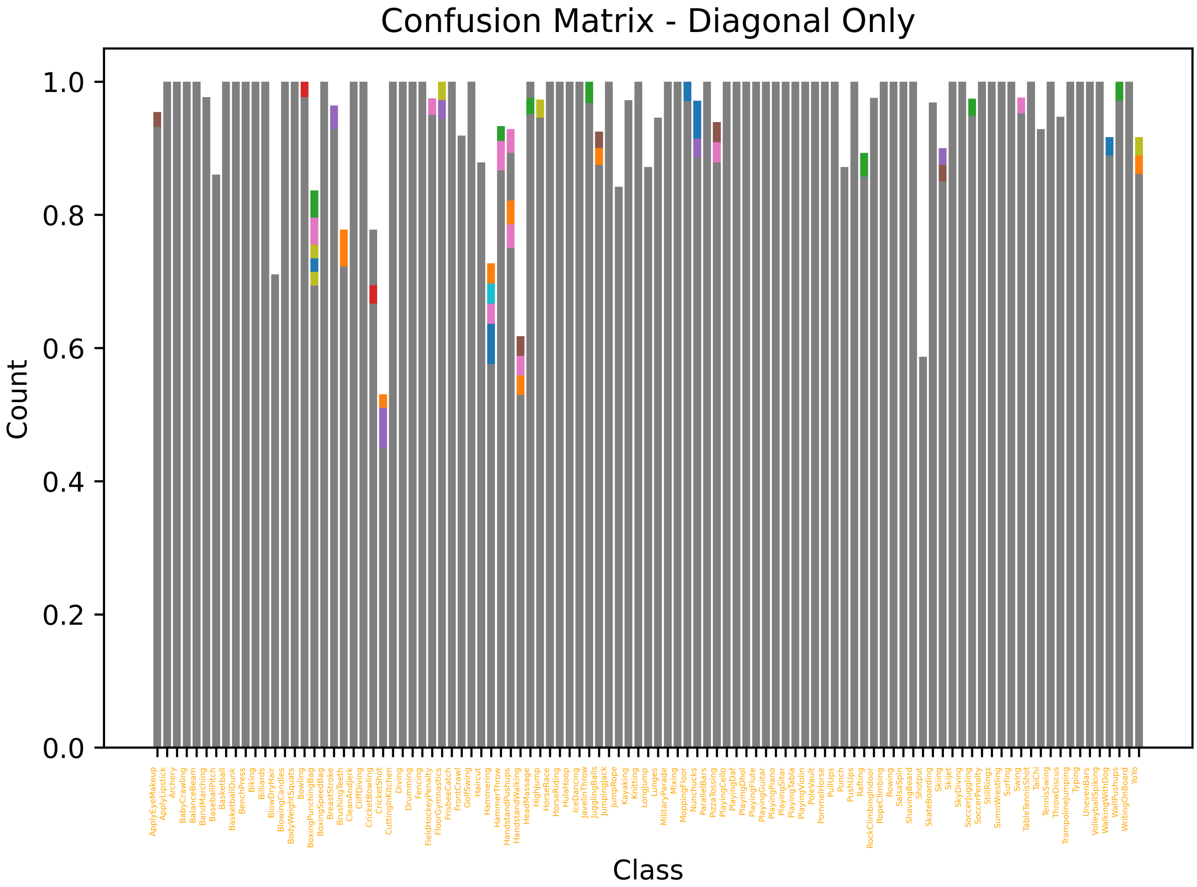

Figure 5: UCF101 confusion matrix prediction on test set

[7] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[9] Xuefan Zha, Wentao Zhu, Lv Xun, Sen Yang, and Ji Liu. Shifted chunk transformer for spatio-temporal representational learning. *Advances in Neural Information Processing Systems*, 34:11384–11396, 2021.

[10] Yanyi Zhang, Xinyu Li, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. Vidtr: Video transformer without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13577–13587, 2021.