COMP411/511: Project Report

January, 20th

Cem Ozan Doğan Azad Aslanlı Tolga Ermiş

1 Introduction

- **Problem Definition**
  The problem we are trying to tackle in our project is action recognition in videos. This task involves identifying and classifying human actions or behaviors from sequences of video frames. To tackle this challenge, we used a transformer architecture as proposed by [8], leveraging multiple input representations in the form of different ”views”, which are essentially different token sizes input to transformer encoders. This allows us to process and learn from multiple viewpoints of a given action, enhancing the accuracy and robustness of recognition.
  We aim to make the model more resource-efficient, by experimenting with different hyperparameters and by integrating mechanisms offering potential advantages.

- **Motivation**
  Action recognition in videos is a critical problem with wide-ranging applications such as human-computer interaction, healthcare and autonomous systems. Accurate recognition of human actions can enhance AI-powered systems. Current solutions have made significant progress, but many are not scalable or efficient, particularly when processing longer or high-resolution videos. By improving the multiview transformer architecture and making it more scalable, our work addresses this gap. We aim not only to enhance action recognition performance but also to optimize the model's ability to work with more limited resources, making it more accessible, and its ability to handle larger datasets and more complex video inputs.

- **Contributions**
  Our initial objective was to evaluate the impact of incorporating a different attention mechanism, specifically Flash Attention, into the architecture to determine whether it could enhance accuracy or accelerate the training process. Flash Attention is a memory-efficient implementation of the attention mechanism that reduces computational overhead by avoiding explicit storage of the attention matrix, making it faster and more scalable than traditional attention [1].
  For this, we had to implement a working baseline model to make modifications on it. However, achieving a functional baseline model proved to be more challenging than anticipated. As a result, we shifted our focus to achieving comparable results using a more scalable model. To accomplish this, the model parameters were adjusted and the dataset was sampled to make training faster with our limited resources.

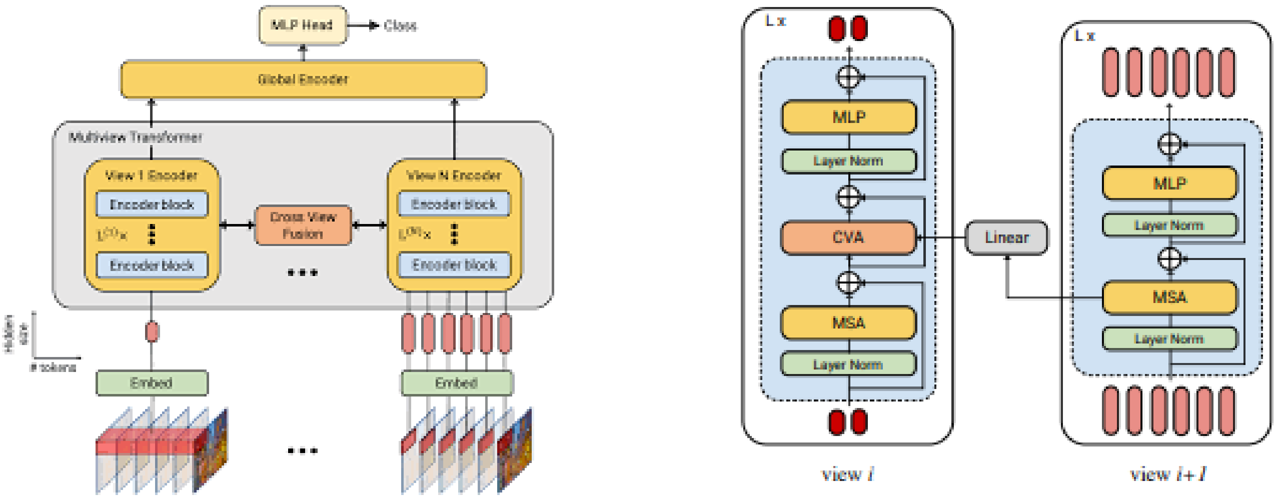
2 Related work

- **Starting Point**
  Identify any baseline or existing work you are using as a foundation. Explain why you chose this work and how it fits with your project. Our foundation work is the Multiview Transformer [8]. After relying on hand-crafted features [5] to encode motion and appearance, CNN and RNN architectures became the preferred methods for video understanding tasks [4]. But the convolution operations can only process a local neighborhood at a time, and are outperformed by transformers in modeling of long range dependencies. After pure transformer architectures gained popularity in computer vision with Vision Transformers [2], some works adopted the transformer architecture for video classification.
  We selected this work as our baseline because it presents a novel approach to the problem by leveraging the very popular transformer architecture. This approach builds upon the existing applications of transformers in video understanding, a field which we believe can greatly benefit from the transformers' ability to capture long-range dependencies and contextual relationships. We believe that this combination offers significant potential to improve the accuracy and efficiency of video analysis, making it a promising solution for action recognition tasks.

- **Related Work**
  There exist recent research that focuses on improving the efficiency of attention mechanisms and transformer architectures. For example, [1] introduces a memory-efficient implementation of the attention mechanism, aiming to improve computational complexity and memory-efficiency. Similarly Linformer [7] uses low-rank projections to reduce the quadratic complexity of self-attention.
  The goal of our project is to enhance the efficiency of the Multiview Transformer model by tuning its hyperparameters and by using such mechanisms.

3 Method

- **Overview**
  We decided to try different layer numbers to test which ones will be the best. Additionally, we tried to give the global encoder different tokens to see the results.



(a) General MVT Architecture

(b) Cross View Attention(CVA) Mechanism

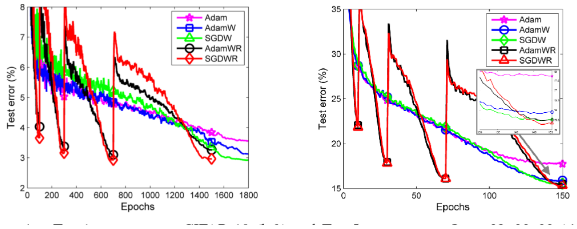
Figure 1: [8]

- **Model Details**

  Our model is a multi-view transformer. From its name we can see that it has many transformer encoder layers. At the first step, we tokenize the videos using tubelets which have the shape (frame_num, height, width). This tokenization is similar to 3D convolution because we get the tubelets with stride equal to their size, so we get non-overlapping patches of 3D tensors and project them to fit the view modules' embedding dimensions. After this step we give the tokens to the each view module. Inside the view modules we have layers of transformer encoders. Transformer encoders follows the structure of the original ViT paper[2]. Additionally, there are cross-attention paths from one view's layer to the other view's layer. Basically it creates a table structure which the right view can inform the left view with its intermediate calculated tokens. These intermediate tokens pass through the attention mechanism and then jiins the left views computations by concatenation. After all views processed their respective tokens, they are projected to the global embedding dimension and concatenated. Global encoder which is basically a many layered transformer encoder process these tokens and gives the resulting output to a basic Multi Layer Perceptron(MLP) to classify the video. There are many models architectures which we have tested in our project: 3 views with 2 layers each, 3 views with 4 layers each, 3 views with 6 layers each, giving all processed tokens to the global encoder, giving only the CLS tokens to the global encoder and giving temporal+CLS tokens to the global encoder.

- **Training Details**

  We used AdamW optimizer with cross-entropy loss to differentiate the video classes. We used AdamW optimizer because it is used in many state-of-the art deep learning models especially in transformers. It separates the weight decay from the gradient calculations and helps the models converge to a minimum faster. We used cross-entropy with softmax because it is widely used in classification tasks[6].



(a) Different Learning Rate Algorithms

Figure 2: [6]

- **What's New**

  In our project we tried many different tokenization strategies. For example, we tried to get the mean of all tokens to get a better summary of the video, we created CLS tokens and only passed these tokens to the global encoder, we used both temporal and CLS tokens to better inform the global encoder. Additionally, we tried to train the models without any pretraining. The convergence was much slower in the non-pretrained models.

4 Results

- **Datasets and Metrics**

  The main dataset that we use for training process is Something-Something V2 [3] dataset which is consist of 3 or 4 second videos of simple motions of human interactions with different objects. In total around 200,000 video included in dataset and because of that same/similar backgrounds and objects appear in different classes this dataset is very challenging.

The primary evaluation metric is accuracy which measures ratio of correct predictions to total predictions.

- **Evaluation Protocol**
  For running experiments different dataset portions are tested. One of very starting points is using 20,000 random data points and using them. Something-Something V2 dataset itself consists of train, validation and test parts, and we use each train and test parts accordingly. Also data points from the top-10 most common classes in the training dataset are used in most models. The reason of choosing these approach is that training through all dataset takes so much time and GPU. But as a finisher, model trained with whole dataset after considering that model started to over-fit with top-10 classes and in general, transformers give much better results with larger dataset if they are good structured enough for handling large data.
  Despite that different hyper-parameters and structural differences are used for seeing effect of such changes in the performance of model, common structure is used in all experiments. Total number of epochs is predefined, and at the end of each epoch, train and test accuracies are monitored. Every time if test accuracy is higher than previous best new model is saved. Also limit epoch used for saving model in some extra-ordinary situations such that credit for GPU is not enough, and early stopping patience is used for stopping training if accuracy does not improves for some time.
  Also for reducing model size and making it easy to fit in GPU we use Gradient Accumulation. By this method we reduce our actual batch size while preserving our effective batch size with help of accumulation steps. With this method we also get less noisy gradient updates.
  Because of that model is learning very slow, we also add pre-training part to our model. For each of 3 views weights from ViT-Base, ViT-Small, ViT-Tiny‘s layers taken correspondingly and used. By this way faster convergence and better performance is targeted.

- **Results**
  Our model continues the training process right now. Some results which are not very successful and caused by some mistakes in parameter and structure selection is mentioned in next section. The model which we are going to present in meeting is best version of our MultiView Transformer which includes 6 layer with CVAs in [1,2,3,4] layers (zero based indexing), pretrained with ViT Base, Small, Tiny weights and uses Gradient Accumulation. With this model we are planning to beat the baseline of project.

- **Ablations and Experiment Conclusions**
  Different parameter and structures are tried and generally over-fitting is main issue throughout the processes.
  One of the first mistakes is using 20000 random data points for training. But results are dramatically low: after 30 epoch testing accuracy is only 15 percent and testing accuracy is nearly 0.03 percent which is under random. At this point we realize that 20000 random data points is not effective because of there are 174 total classes and some classes are underrepresented in train dataset.

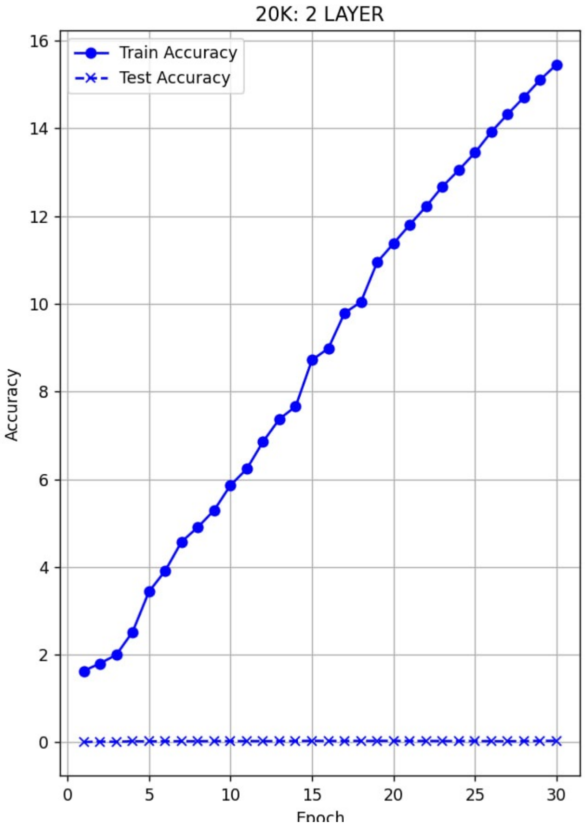


Figure 3: Training results with 20K dataset

Also in very first tries some another important problems about model is seen. For example in our first try with top-10 dataset, 2 layer testing accuracy is increased by time and reach 100 percent at epoch 40, but testing accuracy stayed at 30 percent. We first tried to increasing dropout for solving these problem but it did not help. Then we realize that problem is because of our temporal tube sizes which are (16, 32, 64) respectively, but our videos include at most 48 frames and most of videos are not split which causes these over-fitting and recognizing actual patterns.
Also making changes in layer number is tried which is targeted to finding optimal mini model which best fits top-10 dataset. Therefore a model tested with 4 layer with Cross View Attentions in each layer and 6 layer with Cross View Attentions in layers [1, 2, 3, 4] (zero indexing is used and CVA not used in very first layer and last layer), but either of models showed low improving in train and nearly zero improvement in test after some points.

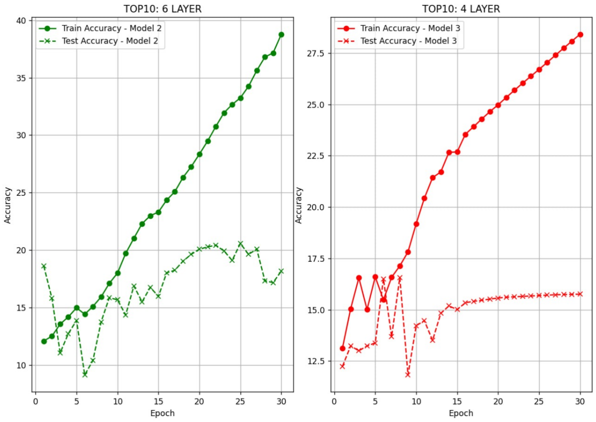


Figure 4: Top-10 class training with 4 and 6 layer views

5 Conclusion

In conclusion, our project explored the performance of Multiview Transformers for action recognition in videos by systematically experimenting with different hyperparameter configurations. By tuning parameters such as learning rate, batch size, number of attention heads, view layer number, we gained valuable insights into their impact on the functioning of the Multiview Transformer.
Throughout this project, we learned how the hyperparameters of attention mechanisms and transformer architectures affect accuracy and training. Our main challenge when working with videos was fitting models and batches within limited resources while managing long training times. To address this, we used techniques like gradient accumulation and distributed training. We also worked on video data preprocessing and sampling to make training feasible given our resource constraints.
Future work would continue to try to optimize hyperparameters, and also focus on integrating different mechanisms into the Multiview Transformer architecture.

6 Acknowledgments

Cem Ozan Doğan: Literature Review, Dataset Preparation, Model Design and Implementation, Training, Reporting.
Tolga Ermiş: Literature Review, Model Design and Implementation, Training, Reporting.
Azad Aslanlı: Literature Review, Model Design and Implementation, Training, Reporting.

References

[1] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

[2] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[3] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017.

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

[5] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference*, pages 275–1. British Machine Vision Association, 2008.

[6] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2019.

[7] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[8] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3333–3343, 2022.