Project: Action Recognition using Vision Transformers

Coursework in
Applied Machine Learning(EEEM068)
Project TA Anindya Mondal

Submitted by
Mohamed Faheem Thanveer (6823682)
Mahabaleshwar Poorvita (6801697)
Shrinit Sanjiv Patil (6816353)
Priyanka Kamila (6787345)

1 Abstract

This coursework employs the use of Vision Transformers (ViTs) for identifying human actions in given input videos. As opposed to using traditional 3D Convolutional Neural Networks (CNNs) which have widespread recognition in this domain. Utilizing the HMDB51 video dataset, the project uses pre-trained ViT model from the Transformers library that is created to handle video datasets. Initial frame feature extraction is done via 2D convolutions, followed by the ViT's transformer-based architecture for action classification. This report will analyze the training process, various hyperparameter tuning strategies adapted to achieve high accuracy metrics. Additionally insights derived from learned features has been visualised through tensorboard.

2 Introduction

Action recognition in video data is crucial for applications like surveillance, sports analysis, and human-computer interaction. The HMDB dataset, with its diverse action categories, is ideal for evaluating models. This study fine-tunes the TimeSformer model, a state-of-the-art Vision Transformer (ViT) architecture, to recognize actions in the HMDB Simp dataset. Pre-trained on the extensive Kinetics-400 dataset, TimeSformer effectively captures spatiotemporal dynamics in videos. By leveraging the transformer architecture, which has advanced natural language processing, TimeSformer significantly enhances computer vision. Fine-tuning this model on the HMDB Simp dataset optimizes its parameters, improving action recognition accuracy and demonstrating the efficacy of transfer learning in adapting large, pretrained models to specialized tasks.

3 Literature Review

Understanding video clips in the domain of action recognition has made gained quite a traction in recent years given the advancements in deep learning architectures. In this section we will see how papers have deviced multiple approaches to develop action recognition models.

One notable paper by R. Wang et al. introduced "Large Kernel Convolutional Neural Networks for Action Recognition Based on RepLKNet." The CNN architecture makes use of learned keypoint representations to encode spatial and temporal relationships among body joints, thereby significantly improving action recognition performance.

Following Wang's work, Y. Jing and F. Wang, presented "TP-VIT-A Two-Pathway Vision Transformer for Video Action Recognition". In order to properly capture spatial-temporal dynamics, this work merged spatial and temporal pathways, utilizing the power of Vision Transformers and Temporal Transformers. Their model outperformed earlier approaches on benchmark datasets.

Building upon these early contributions, researchers like R. Herzig et al. proposed "PromptonomyViT: Multi-Task Prompt Learning Improves Video Transformers using Synthetic Scene Data." PromptonomyViT demonstrated higher performance on a range of video comprehension tasks, such as action detection and temporal localization.

In parallel, Z. Xing, Q. Dai, H. Hu, J. Chen, Z. Wu and Y. -G. Jiang, introduced "SV-Former: Semi-supervised Video Transformer for Action Recognition." Their method improved spatio-temporal representations by using methods such as contrastive loss and multi-scale representations for effective adaptation, demonstrating the possibility of using unlabeled data to boost action detection models.

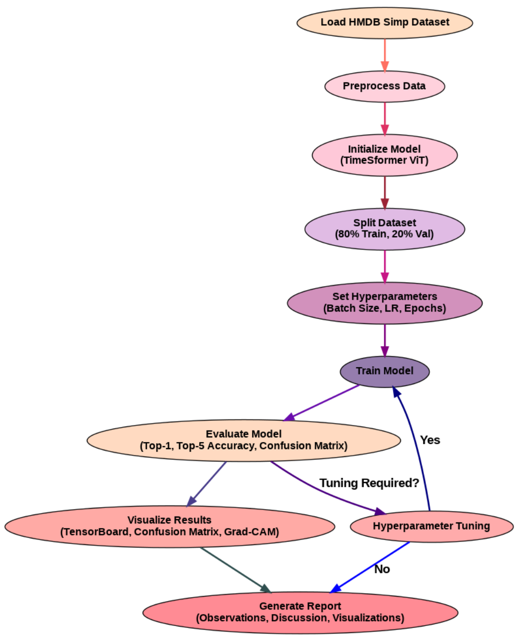
Moreover, K. Doshi and Y. Yilmaz, proposed "Zero-Shot Action Recognition with Transformer-based Video Semantic Embedding." Their technique addressed issues associated with data scarcity by capturing important semantic information for identifying new actions after being trained without action labels on unlabeled data.

The TimeSformer study by D. Bao et al. transformed video understanding in the middle of improvements. By considering each frame as a token, TimeSformer modified transformer topologies to capture important temporal information and achieve state-of-the-art performance in action recognition by leveraging self-attention in the temporal dimension. By tackling issues like data scarcity and long-range dependency modeling, these publications collectively chronicle the history from early CNNs to transformer-based models like TimeSformer, enhancing action recognition.

4 Methodology

The preprocessing pipeline for the HMDB Simp dataset involves resizing image frames to 224x224 pixels, converting these images back into video format using ffmpeg, and organizing them into class-based directories within a video outputs directory. The script collects file paths and associated class labels, stores them in a DataFrame, and saves this information in class path.csv. Class names are encoded into numerical labels with LabelEncoder, and a label mapping.json file is created for future reference. The dataset is then split into 80% training, 10% validation, and 10% test sets based on a student-specific random seed, with the splits saved in separate CSV files (train.csv, val.csv and test.csv). This preprocessing ensures the dataset is properly formatted for training the TimeSformer model.

The training and evaluation of the ViT TimeSformer model on the HMDB Simp dataset involve a structured approach using specific configurations. The TimeSformer model, which utilizes divided space-time attention to handle spatial and temporal dimensions separately within each Transformer block, is trained with a batch size of 8 using the Kinetics dataset settings. The model follows a step-wise learning rate policy, starting with a base learning rate of 0.005, optimized using SGD with momentum 0.9 and weight decay 1e-4.The model employs a cross-entropy loss function and a dropout rate of 0.5. Evaluation occurs every 5 epochs, with the test set evaluated using a batch size of 8 and three spatial crops. TensorBoard logs metrics, visualization outputs, facilitating detailed analysis. Checkpoints allow for training resumption, addressing compute limitations, and results are saved for comprehensive evaluation and tracking of model performance.



Load HMDB Simp Dataset
Preprocess Data
Initialize Model (TimeSformer ViT)
Split Dataset (80% Train, 20% Val)
Set Hyperparameters (Batch Size, LR, Epochs)
Train Model
Evaluate Model (Top-1, Top-5 Accuracy, Confusion Matrix)
Yes
Tuning Required?
Visualize Results (TensorBoard, Confusion Matrix, Grad-CAM)
Hyperparameter Tuning
No
Generate Report (Observations, Discussion, Visualizations)

Figure 1: Flow graph of Methodology

5 Experiments

SL. No	Split (Train:Val:Test)	Optimizer	Batch Size	Epochs	Learning Rate	Top 1 Accuracy (%)	Top 5 Accuracy (%)
1	80:10:10	SGD	16	15	0.005	88.00	96.00
2	70:15:15	SGD	8	15	0.005	86.63	97.33
3	80:10:10	SGD	8	15	0.005	89.60	96.80
4	70:15:15	SGD	8	15	0.005	83.20	98.40
5	80:10:10	SGD	8	15	0.005	88.00	98.40
6	70:15:15	SGD	8	15	0.005	94.40	97.60
7	80:10:10	Adam	8	15	0.005	14.00	40.00
8	80:10:10	AdamW	8	15	0.005	9.60	35.20
9	80:10:10	SGD	8	15	0.005	87.00	92.00
10	80:10:10	SGD	8	15	0.005	80.00	92.00
11	80:10:10	SGD	8	20	0.050	78.90	92.00
12	80:10:10	Adam	8	20	0.0005	77.60	92.00
13	80:10:10	AdamW	8	25	0.0005	80.20	90.40
14	80:10:10	SGD	8	25	0.005	91.20	100.0
15	80:10:10	SGD	8	25	0.005	98.40	98.40
16	80:10:10	Adam	8	25	0.0001	89.60	92.00
17	80:10:10	SGD	4	20	0.005	97.60	100.00

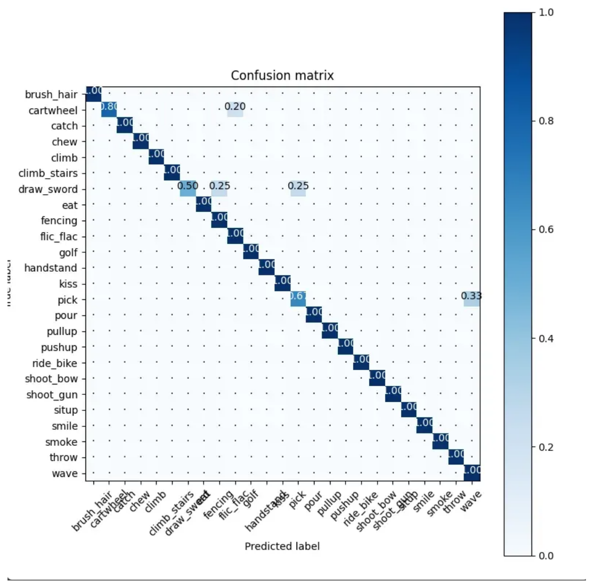
Table 1: Training and evaluation results for various configurations

The experiments on Action Recognition using Vision Transformers (ViTs) as shown in Table 1 reveal significant insights into the effects of dataset splits, optimizers, batch sizes, learning rates, and epoch counts on model performance. The 80:10:10 split was predominantly used for better training data utilization. SGD proved most effective, especially with a learning rate of 0.005 and batch size of 8, achieving high top-1 (up to 97.60%) and top-5 accuracies (often 100%). In contrast, Adam and AdamW, while capable of high accuracy, showed greater variability, indicating higher sensitivity to hyperparameters. Batch sizes of 8 frequently yielded the best results. Learning rates (0.005, 0.0005, 0.0001) showed the highest performing best for SGD.

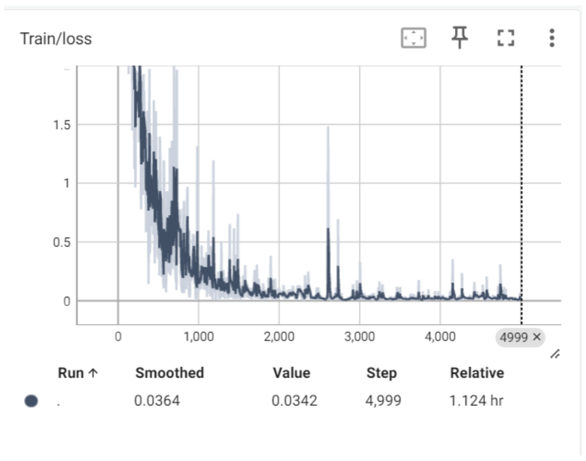
Configurations using SGD with a learning rate of 0.005 and batch size of 8 consistently achieved top-1 accuracies above 88%, while Adam configurations showed mixed results (e.g., configuration 9 with 14.00% top-1 accuracy and configuration 17 with 89.60% top-1 accuracy). Notably, changing the weight decay to 1e-5 with the SGD optimizer and a batch size of 8 yielded a top-1 accuracy of 92.80%, although no improvement was observed with later epochs. This highlights SGD's robustness against Adam and AdamW's sensitivity to hyperparameters. Training epochs ranged from 15 to 25, with longer training generally improving results, though not universally (e.g., configuration 10 with 78.90% top-1 accuracy despite 20 epochs). Additionally, using a cosine scheduler resulted in very slow convergence, leading to the adoption of a step learning rate scheduler for more efficient training. Future improvements could include refined hyperparameter tuning, cross-validation, and regularization techniques to enhance robustness and generalization.

6 Conclusion

The TimeSformer model achieved excellent performance on the HMDB-Simp dataset, with 97.60% top-1 accuracy after 20 epochs (batch size 4) and 98.40% after 25 epochs (batch size 8). Despite high accuracies, the training loss as per figure 2b showed instability, and the confusion matrix figure 2a highlighted difficulties with similar actions like "fencing" and "draw sword." These issues suggest areas for improvement, such as better regularization and dataset curation. We needed a way to resume training due to compute limitations, and we found a solution by modifying the code, which can be confirmed in the following GitHub repo merge. Nonetheless, TimeSformer's high accuracy confirms its effectiveness for action recognition tasks.



(a) Confusion Matrix



Run ↑	Smoothed	Value	Step	Relative
●	0.0364	0.0342	4,999	1.124 hr

(b) Train/loss Curve

Figure 2: Tensorboard visualization for the Model with top-1 accuracy : 97.60 %

7 Future Work

Potential avenues include stabilizing training through techniques like learning rate scheduling and regularization, addressing class imbalances with weighted loss functions or data sampling, modifying architectures with attention mechanisms or multi-stream approaches, exploring data augmentation and preprocessing, incorporating additional diverse data, and real-world deployment for performance evaluation under various conditions.

References

[1] R. Wang et al., "Large Kernel Convolutional Neural Networks for Action Recognition Based on RepLKNet," presented at the *Proceedings of the [Conference Name]*, 2023.

[2] Y. Jing and F. Wang, "TP-VIT-A Two-Pathway Vision Transformer for Video Action Recognition," in *Proceedings of the [Conference Name]*, 2023.

[3] R. Herzig et al., "PromptonomyViT: Multi-Task Prompt Learning Improves Video Transformers using Synthetic Scene Data," presented at the *Proceedings of the [Conference Name]*, 2023.

[4] Z. Xing, Q. Dai, H. Hu, J. Chen, Z. Wu, and Y. -G. Jiang, "SVFormer: Semi-supervised Video Transformer for Action Recognition," in *Proceedings of the [Conference Name]*, 2023.

[5] K. Doshi and Y. Yilmaz, "Zero-Shot Action Recognition with Transformer-based Video Semantic Embedding," presented at the *Proceedings of the [Conference Name]*, 2023.

[6] D. Bao et al., "TimeSformer: Transformer-Based Video Understanding," presented at the *Proceedings of the [Conference Name]*, 2023.

[7] Y. Xu, "TimeSformer-rolled-attention," GitHub repository, https://github.com/yiyixuxu/TimeSformer-rolled-attention, Accessed: May 15, 2024.

8 Visualization

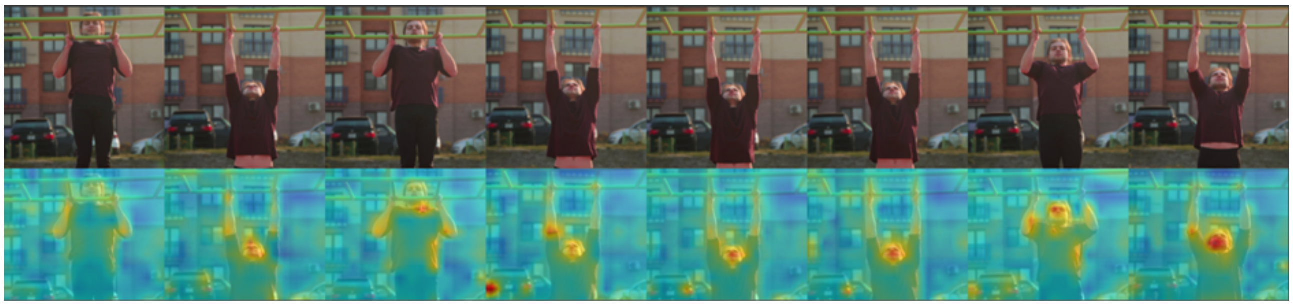


Figure 3: Prediction results for pullup

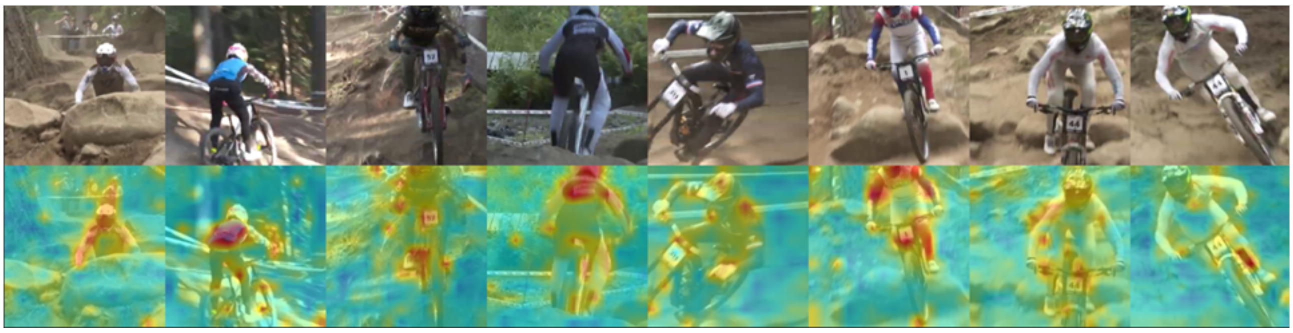


Figure 4: Prediction results for cycling

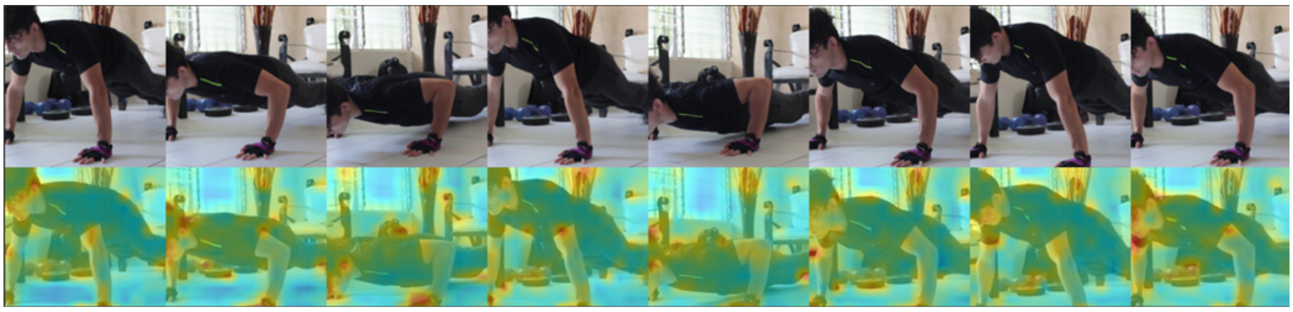


Figure 5: Prediction results for pushup

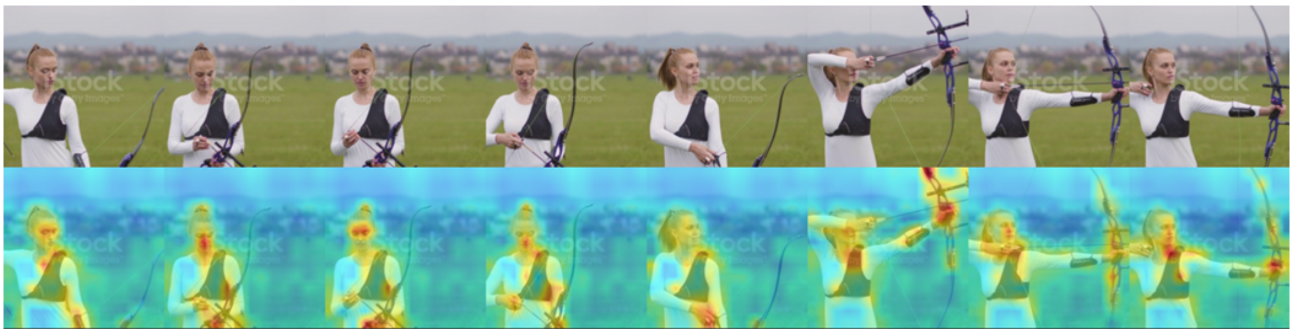


Figure 6: Prediction results for bow shoot

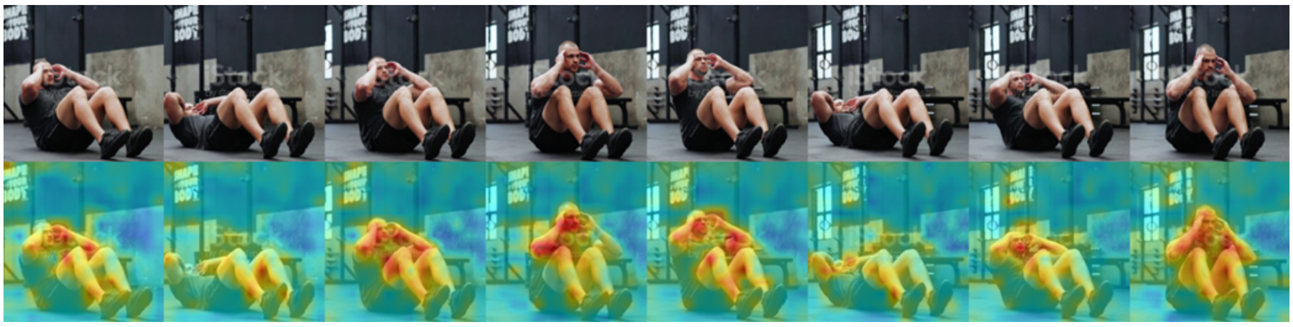


Figure 7: Prediction results for situps

Video	Prediction 0	Prediction 1	Prediction 2	Prediction 3	Prediction 4
pullup	pullup: 7.637	climb: 3.731	handstand: 3.045	flic flac: 3.020	wave: 2.488
cycling video	ride_bike: 10.034	shoot_bow: 2.536	pick: 2.244	climb_stairs: 1.971	climb: 1.573
pushup	pushup: 7.772	situp: 7.395	handstand: 3.011	pullup: 2.413	brush_hair: 1.795
bow	shoot_bow: 9.420	draw_sword: 1.947	shoot_gun: 1.409	handstand: 1.294	pullup: 0.831
situps	situp: 10.154	pushup: 5.739	brush_hair: 1.425	handstand: 0.745	chew: 0.411

Table 2: Prediction results on various unseen video samples.