# *Video Classification and Action detection Using Video Transformer*

Advisor

Prof. Hong Man
Department of Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, NJ-USA
E-mail: hman@stevens.edu

Student

Hinalben Patel
Department of Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, NJ-USA
E-mail: hpate98@stevens.edu

***Abstract*— A transformer-based framework for video categorization and action detection is presented in this research. We abandon a conventional methodology in the video activity recognition that depends on, and create a system that classifies actions by attention to the complete video sequence information, inspired by recent breakthroughs in vision transformers.In this research project, we implement the video transformer model for video classification and action detection in the human activities video dataset of UCF-101.The dataset comparatively huge than the other dataset, so that for this project we are just used five class for prediction. We analyzed that the model perform very well, it gives the testing accuracy of 87.05% relatively very good than other models. However, the model gives the wrong prediction result for some classes such as tennis swing and cricket shot, because of some video quality of the dataset.**

***Index terms*—***Video Classification, Action Recognition, Vision Transformer, Spatial - temporal attention*

## I. INTRODUCTION

Video classification involves labeling a video based on its frames. A excellent video level classifier not only classifies frames accurately, and also summarizes the whole film based on its characteristics and comments. In Computer Vision, We Know that detecting Key Frames in a videos is a most challenging part for many application such as Video Summarization, Visual simultaneous localization and mapping and Action detection. The Project idea is to detecting the activities in videos using Video Transformer(ViT). A video is often three-dimensional in nature. Instead, than just a sequence of frames, it is a mix of the audio, motion, and time series dimensions as well as a time series of frames. This requires the use of transformer models rather than neural networks models to exceptionally large amount of video data to be easily browsed. Nowadays Vision Transformers are very popular and powerful, than the CNN(Convolution Neural Network) for image classification, object detection and video classification. In this project we are trying to implement the model for video classification to get better accuracy than the CNN models.

Mingze Xu and et al., They introduce Long Short-term TRansformer (LSTR), a temporal modeling technique for online action detection that models extended sequence data using a long- and short-term memory mechanism. It comprised of an LSTR encoder that dynamically leverages coarse-scale historical information from an extended temporal window (e.g., 2048 frames spanning up to 8 minutes) and an LSTR decoder that concentrates on a short time window (e.g., 32 frames spanning 8 seconds) to model the data's fine-scale characteristics. LSTR, in comparison to previous work, presents an effective and efficient strategy for modeling lengthy films with less heuristics, which is proven by substantial empirical investigation. LSTR outperforms the competition in three typical online action detection benchmarks: THUMOS'14, TVSeries, and HACS Segment[2].

J. Lee and et al., They describe the challenge of tailing detection from films as an anomaly identification problem, with the purpose of detecting abnormalities in pedestrian movement patterns (victim and follower). As a result, they offer a modified Time-Series Vision Transformer (TSViT), a technique for detecting anomalies in video, notably tailing detection with a short dataset. The researchers provide a novel method for training TSViT with a minimal dataset by regularizing the prediction model. To accomplish so, they first encode the pedestrians' spatial information into 2D patterns and then transmit these as tokens to the TSViT. They demonstrate through the use of a number of tests that tailing detection on a short dataset using TSViT outperforms common CNN-based designs, since CNN architectures tend to overfit with a small dataset of time-series pictures. They also demonstrate that when employing time-series pictures, the performance of CNN-based design steadily decreases when network depth is raised in order to enhance capacity. On the other hand, a decreasing number of heads in the Vision Transformer design performs well on time-series pictures, and the performance improves as the image input resolution increases. The experimental findings show that the TSViT outperforms the handmade rule-based technique and the CNN-based method for tailing detection. TSViT may be used to identify video anomalies in a variety of applications, even with a limited dataset[3].

Chiara Plizzari and et al.,Researchers propose a new Spatial-Temporal Transformer network (ST-TR) that simulates joint dependencies using the Transformer self-attention operator in this paper. A Spatial Self-Attention module (SSA) is utilized in their ST-TR model to comprehend intra-frame interactions between distinct body parts, while a Temporal Self-Attention module (TSA) is used to represent inter-frame correlations. The two are merged in a two-stream network, and its performance on three large amounts of data, NTU-RGB+D 60, NTU-RGB+D 120, and Kinetics Skeleton 400, consistently improves backbone results. When utilizing joint coordinates as input, the proposed ST-TR achieves state-of-the-art performance on all datasets, and performs on par with state-of-the-art when adding bone information[4].

Sanchez-Caballero and et al., This study offers a complete method for real-time human activity identification from raw depth picture sequences. The approach is based on 3DFCNN, a 3D fully convolutional neural network that automatically encodes

spatio-temporal patterns from raw depth data. They explained how 3D-CNN can classify activities based on the spatial and temporal encoded information of depth sequences. Because this data cannot be used to identify people's identities, the use of depth data enables that action recognition is carried out while safeguarding their privacy. The suggested 3DFCNN has been designed to achieve excellent accuracy while operating in real-time.It has been analyzed and compared with various state-of-the-art systems in three widely used public datasets with diverse features, indicating that 3DFCNN surpasses all the non-DNN-based state-of-the-art techniques with a maximum accuracy of 83.6% and produces results that are equivalent to the DNN-based approaches, while keeping a substantially lower computing cost of 1.09 seconds, greatly increasing its usability in real-world environments[5].

Wendi Li and et al., They present a unique framework in this research that may pick the discriminative component in the spatial dimension while enriching the modeling action of motion in the temporal dimension. They use part selection inside clips and take bidirectional temporal information into account when modeling the temporal pattern using various layers of a long short-term memory framework that can learn compositional representations in space and time. Their findings on the standard benchmarks UCF101 and HMDB51 indicate that the suggested architecture provides cutting-edge performance[6].

From the several studies, most of the researchers were used deep learning networks such as CNN + LSTM, RNN(Recurrent Neural Network), 3DFCNN and Spatial-Temporal Transformer network (ST-TR). Our goal is to develop a model to get minimum and efficient Key-frames to detect actions by using video transformer to classify the videos. Resent studies says that the Vision transformers are most efficient than the Convolution networks in the computer vision task[1]. We will implement on both method convolution networks with LSTM and Video vision transformer and will compare the accuracy result to get better understand of the proposed model.

The rest of the paper is structured as follows. In Section 2, describe the problem statement and give the overall explanation of the project idea. Section 3 is the Solutions for the proposed idea, It describes the data sets, different feature sets and the networks and methods used in Proposed Model, and the detailed explanations of numerical results and analysis and comparisons with other model explained in section 4. Finally, The section 5 is concluded the project idea and research.

## II. PROBLEM STATEMENT

Since AlexNet, approaches based on deep convolutional neural networks have improved the state-of-the-art for several common datasets including vision difficulties.In sequence-to-sequence modeling (e.g., natural language processing), the most popular design is the transformer, which does not employ convolutions but is based on multi-headed self-attention. This technique is very useful for modeling long-distance relationships and enables the model to consider each element in the input sequence. This is in sharp contrast to convolutions, in which the "receptive field" is constrained and develops linearly with network depth[1]. Recently, the success of attention-based models in NLP(Natural language processing) has encouraged efforts in computer vision to include transformers into CNNs, as well as some attempts to totally replace convolutions. With the Vision Transformer (ViT), however, a pure-transformer-based design has lately beaten its convolutional equivalents in picture categorization.

In this paper, the researcher address the topic of video classification and action detection, which involves extracting key-frames from a dataset and producing an output consisting of some meaningful key-frames that predict the valuable information about the video. The problem can be solved by using the video transformer to get the key-frames from input videos. From the past research, they are using many deep learning networks to minimize this problem such as CNN, LSTM and LRCN.

There have been various studies published which used convolution networks to recognize objects in input data, but this study attempts to employ video transformer. Furthermore, labeling is a difficult process for large-scale datasets; to mitigate this issue, we utilized labelled data. We utilized using UCF-101 dataset, that comprises 101 Human action classes based on their location in the Videos.

## III. OUR SOLUTIONS

For train the our model, for the input data, we are using UCF-101 dataset to optimize our results. The data will used in extension of .avi file, with including video and audio. After the reading of the data, we preprocess the data, the dataset is very big it contains 101 class of human activity, so we are just including only five class such as ‘CricketShot’, 'PlayingCello', 'Punch', ‘ShavingBeard' and ‘TennisSwing’ and extract the features from the frames.After that the transformer to embedding position in all the keyframes in the dataset and applied it to the subclassed layer of the Video transformer.

### *A. Description of Dataset:*

The dataset we are using for this project is UCF101 dataset, It is most popular dataset for Human Action Recognization and downloaded from the https://www.crcv.ucf.edu/data/UCF101.php. It includes 101 action courses, more than 13,000 clips, and 27 hours of video data. The database contains actual user-uploaded films with camera movement and a chaotic backdrop[15]. Human-Object Interaction, Body-Motion Only, Human-Human Interaction, Playing Musical Instruments, and Sports are the five kinds of action courses that make up UCF101.For software, we used Jupyter Notebook and the coding done in Python Programming language.

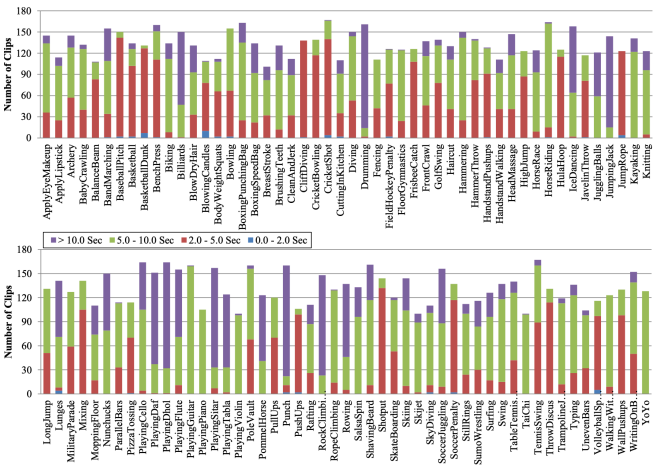



Figure 1: Number of clips per action class[15]

Above figure shows that 101 class names and and how many clips in each classes. After looking above figure we can say that

the dataset is slightly imbalanced, the total video clips in each class is different.

*B. Pre-Precessing:*

As we know that, For pre-processing step, It takes too much time to pre-processed the any data to get efficient result and get the textual information of the any video data. Also, it need powerful GPU to train the large dataset. To overcome this problem, The datasets we used contains only 5 classes such as CricketShot', 'PlayingCello', 'Punch', 'ShavingBeard' and 'TennisSwing. For training we used 594 videos and for testing we used 224 videos. To define hyper parameters we are taking Image size as 128, epochs as 5, maximum sequence length is 20 and 1024 number of features included.

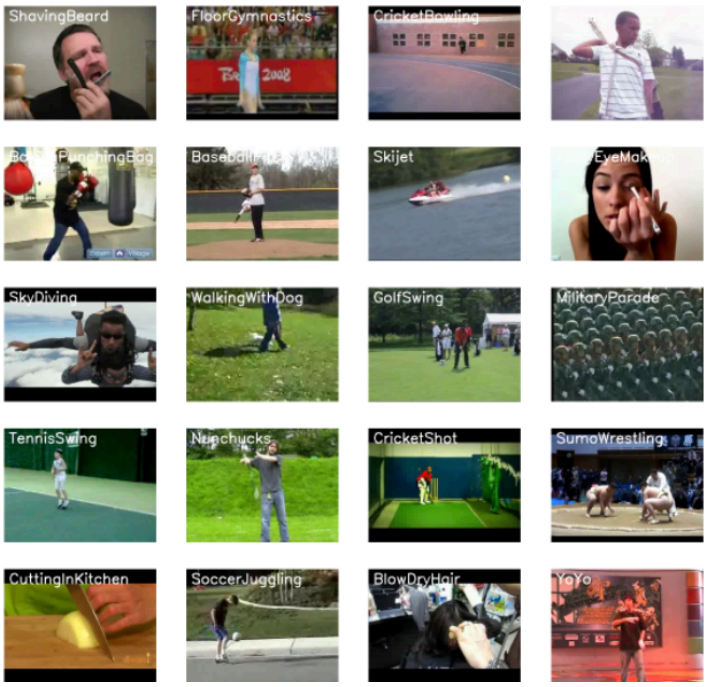



Figure 2: Plot of Pre-processed dataset

After resizing and labeled it the dataset looks like as above, the above plot choose randomly 20 classes from UCF-101 dataset.

```
train_df = pd.read_csv("train.csv")
test_df = pd.read_csv("test.csv")

print(f"Total videos for training: {len(train_df)}")
print(f"Total videos for testing: {len(test_df)}")

train_df.sample(10)
```

```
Total videos for training: 594
Total videos for testing: 224
```

| | video_name | tag |
|---|---|---|
| 330 | v_Punch_g21_c07.avi | Punch |
| 78 | v_CricketShot_g19_c05.avi | CricketShot |
| 496 | v_TennisSwing_g10_c07.avi | TennisSwing |
| 416 | v_ShavingBeard_g17_c02.avi | ShavingBeard |
| 506 | v_TennisSwing_g12_c03.avi | TennisSwing |
| 542 | v_TennisSwing_g17_c04.avi | TennisSwing |
| 283 | v_Punch_g14_c05.avi | Punch |
| 472 | v_ShavingBeard_g25_c02.avi | ShavingBeard |
| 311 | v_Punch_g18_c06.avi | Punch |
| 474 | v_ShavingBeard_g25_c04.avi | ShavingBeard |

Figure 3: Top view of the train dataset

The Figure(3) shows that the head of the training data, also it print out the total video of 594 in training data frame, and the testing data frame contains 224 videos.

*C. Video Transformers Model:*

Transformer has shown outstanding performance in a variety of natural language processing tasks. Recently, a large number of experts sought to investigate the advantages of transformer-based models in problems involving computer vision. A sequence of patches is received as input by the Vision Transformer, which is obtained by tokenizing the input image X ∈ R(HxWxC) into n flattened 2D patches of size p1xp2x C pixels, where H, W, and C are the height, width, and number of channels of the input image, (p1xp2) is the patch size, and n is the number of patches, i.e.,n = H/p1 x W/ p2. Each patch is then projected with a linear layer to D hidden dimensions, and this process is repeated for each patch[2]. Fixed or learnable position embeddings are added to the patch embeddings as an input to the Transformer encoder in order to maintain the relative spatial relationship between the patches[10].

For the classification purpose, a trainable vector (i.e. class token) is attached towards the training data of the patch tokens and then sent through the Transformer encoder to fulfill the classification task. A classification head is then appended to the result of the Transformer encoder corresponding to the class token, and the transformation process is completed. The classification head is built as a single input layer that maps the class embeddings to the class labels in the classification system.

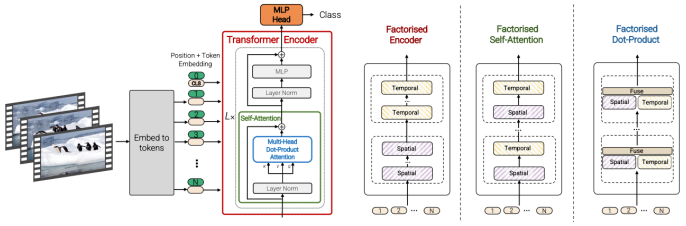



Figure 4: The architecture of the Vision Transformer[1]

The pure-transformer design shown above was motivated by the recent success of such models for images. Several model versions that divide distinct components of the transformer encoder across the spatial- and space - time are used to successfully handle a large number of spatio-temporal tokens. These factorizations correlate too varied attention patterns across space and time, as seen on the right.

**Spatio-temporal attention:**

This model simply passes through the transformers encoder any spatiotemporal tokens derived from the video. Unlike CNN designs, where the receptive field rises linearly with the number of layers, each transformer layer models all pair-wise interactions between all spatiotemporal tokens, modeling long-range interactions throughout the video from the first layer. Multi-Headed Self Attention (MSA) has quadratic complexity in terms of word count since it represents all pairwise interactions. Because the number of tokens rises linearly with the number of input frames in video, this complexity stimulates the development of more efficient systems in the future[1].

**Factorised encoder:**

This type is made up of two distinct transformer encoders. The first, the spatial encoder, exclusively simulates interactions between symbols from the same temporal index. This design represents a "late fusion" of temporal information, and the initial spatial encoder is the same as that used for picture classification. It is therefore identical to CNN architectures, which collect per-frame characteristics before aggregating them into a final representation and categorizing them. Although this model has more transformer layers (and hence more parameters) than the Spatio-temporal attention model, it needs less floating point operations (FLOPs) since the two distinct transformer blocks have a complexity of $O((n_h \times n_w)^2 + n^2t)$ compared to $O((n_t \times n_h \times n_w)^2)$[1].

**Factorised self-attention**

The model has the same computational complexity as the previous models, but with the same amount of variables as the un-factorized Spatio-temporal attention model. In theory, this model's factorization of spatial and temporal dimensions is identical, but it factorize the multi-head dot-product attention operation instead. To be more specific, designers calculate attention weights for each token individually across the spatial and temporal dimensions using multiple heads. To begin, the attention operation for each head is specified as[1]

$$\text{Attention}(Q, K, V) = \text{Softmax}\,(QK^T / \sqrt{d_k})\, V \text{.....(1)}$$

The queries $Q = XW_q$, keys $K = XW_k$, and values $V = XW_v$ in self-attention are linear projections of the input X with X, Q, K, $V \in R^{N \times d}$.

## IV. NUMERICAL RESULTS AND ANALYSIS

Past studies investigate the Video Transformer is more accurate than the CNN and RNN, In this section we will see the both results for CNN and Video Transformer and analyze it, and compare it with each other. After run Video Transformer model, we got accurate result but in some cases the prediction is not accurate. The testing accuracy result around 87.5%

```
trained_model = run_experiment()

Epoch 1/5
16/16 [==============================] - ETA: 0s - loss: 1.1460 - accuracy: 0.7401
Epoch 1: val_loss improved from inf to 4.80944, saving model to /tmp/video_classifier
16/16 [==============================] - 9s 486ms/step - loss: 1.1460 - accuracy: 0.7
Epoch 2/5
16/16 [==============================] - ETA: 0s - loss: 0.1084 - accuracy: 0.9643
Epoch 2: val_loss improved from 4.80944 to 3.22295, saving model to /tmp/video_classi
16/16 [==============================] - 7s 466ms/step - loss: 0.1084 - accuracy: 0.9
Epoch 3/5
16/16 [==============================] - ETA: 0s - loss: 0.0541 - accuracy: 0.9841
Epoch 3: val_loss improved from 3.22295 to 1.05563, saving model to /tmp/video_classi
16/16 [==============================] - 8s 534ms/step - loss: 0.0541 - accuracy: 0.9
Epoch 4/5
16/16 [==============================] - ETA: 0s - loss: 0.0072 - accuracy: 0.9980
Epoch 4: val_loss did not improve from 1.05563
16/16 [==============================] - 7s 454ms/step - loss: 0.0072 - accuracy: 0.9
Epoch 5/5
16/16 [==============================] - ETA: 0s - loss: 0.0109 - accuracy: 0.9980
Epoch 5: val_loss improved from 1.05563 to 0.41808, saving model to /tmp/video_classi
16/16 [==============================] - 8s 507ms/step - loss: 0.0109 - accuracy: 0.9
7/7 [==============================] - 1s 145ms/step - loss: 0.4838 - accuracy: 0.870
Test accuracy: 87.05%
```

Figure 5: Accuracy result for testing data

Now, we see the prediction results for Video transformer model and analyze the performance of it,

**Good Performance**


**Bad Performance**

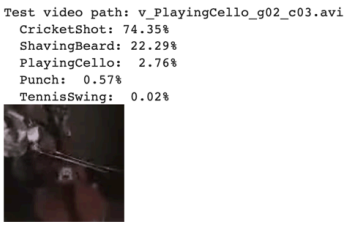



Figure 6: Results for Playing Cello class

Figure(5) shows that the accuracy for the overall performance for the PlayingCello class in the dataset is very good, giving the maximum accuracy of 100 percent, but for another video in this class it predicts 74.35 percent as a cricket shot and only 2.76 percent for the actual class, indicating that it predicted the wrong class. The reason it predicts wrongly is that we can see that the quality of the video in good performance is outstanding and grainy in low performance, therefore it forecasts erroneous class.

**Good Performance:**


Figure 7: Good Predicted results for Tennis Swing class

**Bad Performance**

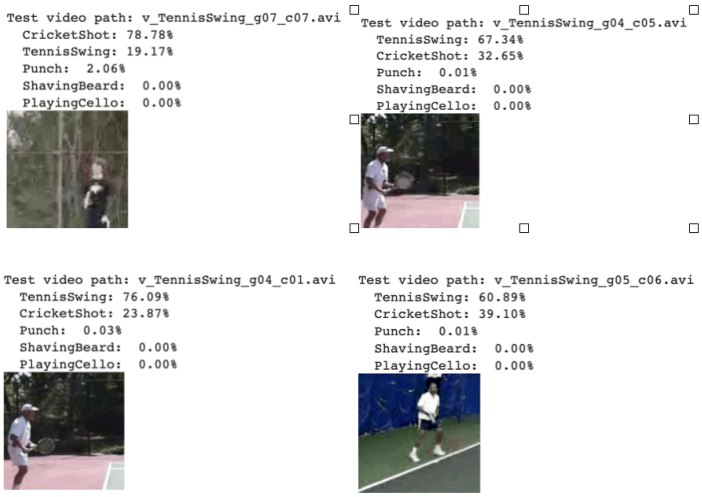



Figure 8: Bad Predicted results for Tennis swing class

In Figure(6), It is the performance for Tennis Swing class, for good results its testing accuracy is around 99%, so that the overall performance is pretty good, for bad results, figure(7) is the minimum accuracy is around 60%, which is not good.


Figure 9: Predicted Results for Punch class

The above figure(8) described as, the overall predicted accuracy for this class is very good, the highest accuracy is 100% and lowest is approximately 88%. From this results we can analyze it the class of punch have a good video dataset, we can see that the quality of video is pretty good, so that it predict correctly.

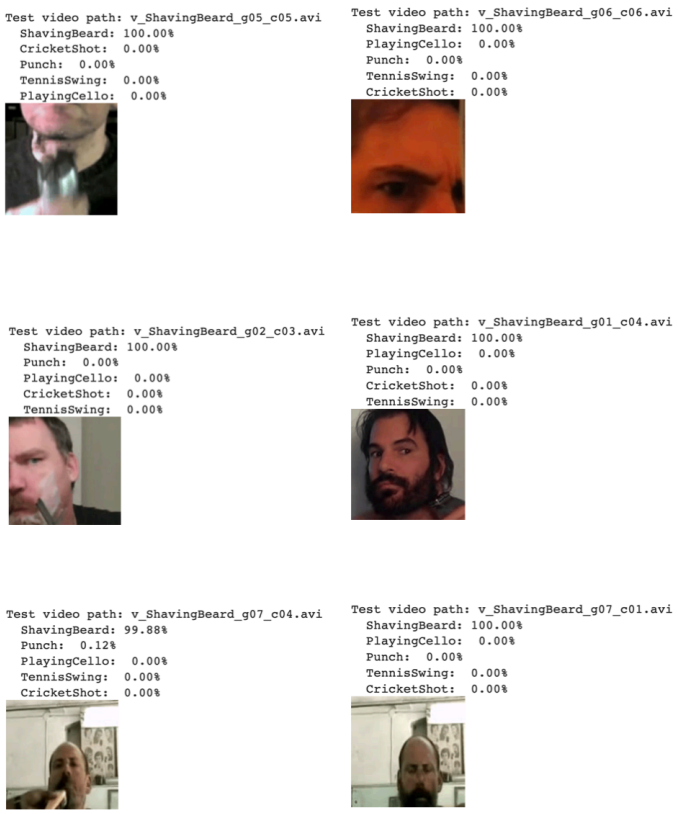



Figure 10: Predicted results for Shaving beard class


Figure 11: Predicted Results for Cricket shot class

From the above results we can analyze it, the Shaving beard class predicted accurately, because we can see in this videos that the frame contains only face features, so the features set compare to another actions, so that it gives around 100% of accurate predicted results.

## Wrong Prediction:


Figure 12: Results for wrong prediction

According to the above figure, several classes, such as tennis swing and playing cello, projected the erroneous class as cricket shot. The cause for this incorrect prediction is the hazy resolution for videos. As previously said, the quality of the video is also an important aspect in obtaining correct findings. Because of fuzzy images or videos, sometimes humans also can not detect actions in videos.

Therefore, we can analyze that the predicted outcomes for several videos show varying degrees of accuracy, from very accurate to somewhat correct to extremely inaccurate. An object's distance and video quality have a significant influence on the model's ability to accurately classify it. The dataset should be trained with good quality data so that it may be put into practice and provide better outcomes. On the other side, we may claim that the performance is not as accurate as it should be since we trained on a small dataset with just five classes. If we utilized this with all 101 classes, the accuracy of the findings will definitely decrease.

Now, we are comparing the transformer model to CNN models.To extract relevant characteristics from the retrieved frames, we may utilize a pre-trained network. The Keras Applications module includes a variety of cutting-edge models that have been pre-trained on the ImageNet-1k dataset. For this, we will use the InceptionV3 model.

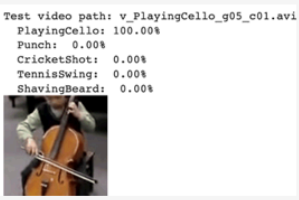
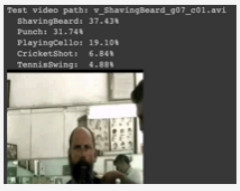
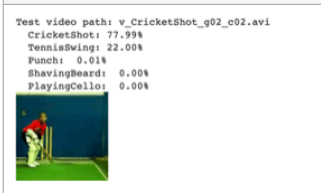
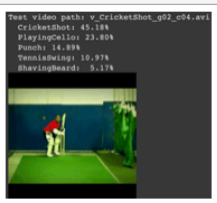
| Video Classification with Video Transformer | Video Classification with CNN-RNN Techniques |
|---|---|
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Table 1: Comparison between Video transformer model and CNN-RNN model

In table 1, we compared the video transformer and CNN-RNN model outcomes. We used the same video clip for each model as testing and examined the prediction accuracy. According to the comparative table, the video transformer model predicts more accurate outcomes than the CNN-RNN model. Furthermore, the CNN-RNN testing accuracy is 76.79 percent, while the video transformer testing accuracy is 87.05 percent. Furthermore, code executive time is reduced in Video transformer.

## V. CONCLUSIONS

We can conclude that the Video Transformer model predict the accurate results with testing accuracy of 87.05%. The model for shaving beard, punch and cricket shot class gives the accurate results for action detection. However, The Class of Tennis swing and Playing cello predict wrong class in some videos, due to the low quality of videos. From this research project we can say that the Video transformer gives the high accurate results.

## References


[1] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić and C. Schmid, "ViViT: A Video Vision Transformer," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 6816-6826, doi: 10.1109/ICCV48922.2021.00676.

[2] Xu, Mingze and Xiong, Yuanjun and Chen, Hao and Li, Xinyu and Xia, Wei and Tu, Zhuowen and Soatto, Stefano," Long Short-Term Transformer for Online Action Detection", 2021, doi:10.48550/ARXIV.2107.03377, arXiv:2107.03377 [cs.CV], https://doi.org/10.48550/arXiv.2107.03377

[3] J. Lee, S. Lee, W. Cho, Z. A. Siddiqui, and U. Park, "Vision Transformer-Based Tailing Detection in Videos," Applied Sciences, vol. 11, no. 24, p. 11591, Dec. 2021, doi: 10.3390/app112411591.

[4] Plizzari, Chiara & Cannici, Marco & Matteucci, Matteo, (2021), "Skeleton-based action recognition via spatial and temporal transformer networks" Computer Vision and Image Understanding, 208-209, pages-103219,10.1016/j.cviu.2021.103219

[5] Sanchez-Caballero, Adrian, Sergio de López Diz, David Fuentes-Jiménez, Cristina Losada-Gutiérrez, Marta Marrón Romera, David Casillas-Pérez and Mohammad Ibrahim Sarker. "3DFCNN: Real-Time Action Recognition using 3D Deep Neural Networks with Raw Depth Information." ArXiv abs/2006.07743 (2022): n. pag.

[6] W. Li, W. Nie and Y. Su, "Human Action Recognition Based on Selected Spatio-Temporal Features via Bidirectional LSTM," in IEEE Access, vol. 6, pp. 44211-44220, 2018, doi: 10.1109/ACCESS.2018.2863943.

[7] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar and L. Fei-Fei, "Large-Scale Video Classification with Convolutional Neural Networks," 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1725-1732, doi: 10.1109/CVPR.2014.223.

[8] Chadha, Aaron and Abbas, Alhabib and Andreopoulos, Yiannis, "Video Classification With CNNs: Using The Codec As A Spatio-Temporal Activity Sensor", Computer Vision and Pattern Recognition (cs.CV), 2017, arXiv:1710.05112 [cs.CV], https://doi.org/10.48550/arXiv.1710.05112.

[9] Zhang, Yanyi and Li, Xinyu and Liu, Chunhui and Shuai, Bing and Zhu, Yi and Brattoli, Biagio and Chen, Hao and Marsic, Ivan and Tighe, Joseph, "VidTr: Video Transformer Without Convolutions", Computer Vision and Pattern Recognition (cs.CV) , 2021, arXiv:2104.11746 [cs.CV], https://doi.org/10.48550/arXiv.2104.11746

[10] D. Chen, X. Wu, J. Dong, Y. He, H. Xue and F. Mao, "Hierarchical Sequence Representation with Graph Network," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 2288-2292, doi: 10.1109/ICASSP40776.2020.9054195

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weis- senborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale." arXiv:2010.11929 [cs.CV], https://doi.org/10.48550/arXiv.2010.11929

[12] Sara Atito, Muhammad Awais, Josef Kittler, "SiT: Self-supervised vision transformer", arXiv:2104.03602v2[cs.CV], November 2021.

[13] X. Yan, S. Z. Gilani, M. Feng, L. Zhang, H. Qin, and A. Mian, "Self-Supervised Learning to Detect Key Frames in Videos," *Sensors*, vol. 20, no. 23, p. 6941, Dec. 2020, doi: 10.3390/s20236941.

[14] T. -C. Hsu, Y. -S. Liao and C. -R. Huang, "Video Summarization With Frame Index Vision Transformer," *2021 17th International Conference on Machine Vision and Applications (MVA)*, 2021, pp. 1-5, doi: 10.23919/MVA51890.2021.9511350.

[15] Soomro, Khurram & Zamir, Amir & Shah, Mubarak. (2012). UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. CoRR.