Video Classification using TimeSformer:

Ankith Dasu
adasu2@kent.edu

1) Summary:

Deep learning is used in video classification to analyze, classify, and track activity contained in visual data sources such as a video stream. We used two main techniques for video classification in this case. The Vision transformer architecture and the Divided Space time attention architecture. A comparison of TimeSformer to the CNN-RNN Hybrid architecture was performed, and it was discovered that the model is faster to train, has significantly higher test efficiency, and can be applied to much longer clips. The accuracy of the model using TimeSfomer was higher, and the other details have been discussed here.

2) Understanding the Problem:

Here, we get a clarity on why using TimeSformer, compared to the traditional CNN-RNN approach is better in terms of accuracy, test, and training results. Video classification is similar to image classification, in that the algorithm uses feature extractors, such as convolutional neural networks (CNNs) and Transformers to extract feature descriptors from a sequence of images and then classify them into categories. The Dataset UCF11(Action recognition dataset) has been used for this model.
So using few techniques, we have seen how TimeSformer has performed extremely well on the video lengths which are longer(>1Minute) and also the lesser training time for the data. It is a convolution-free approach to video classification built exclusively on self-attention over space and time.

3) Techniques:

To implement a proper video classification using TimeSformer Vision Transformer, few techniques have been used in the code process.

- Divided Space time attention(T+S)

  Generally, Attention models are neural networks that focus on specific aspects of a complex sequential input, one at a time until the entire dataset is categorized. In “divided attention”, temporal attention and spatial attention are separately applied within each block, and it leads to the best video classification accuracy. Temporal attention and spatial attention are separately applied one after the other. This attention technique uses all the patches of the current frame and patches at the same position of the adjacent frames.

- Feedforward Mechanism

In addition to the attention mechanism, the TimeSformer includes a full automated Feedforward network. In this architecture, the Feedforward mechanism is the weights that is trained during training and the exact same matrix is applied to each respective token position. Since it is applied without any communication with or inference by other token positions it is a highly parallelizable part of the model.

- Positional Embedding

  The importance of positional embedding here is To investigate the importance of learned spatiotemporal positional embeddings, I have tried few tests with a few variants of TimeSformer. Using No Positional Embedding, Using Space only and Using Space time attention

- Regularization

Large scale pre-training or strong regularization is applied for the training data. Drop outs are used so as to regularize the model.

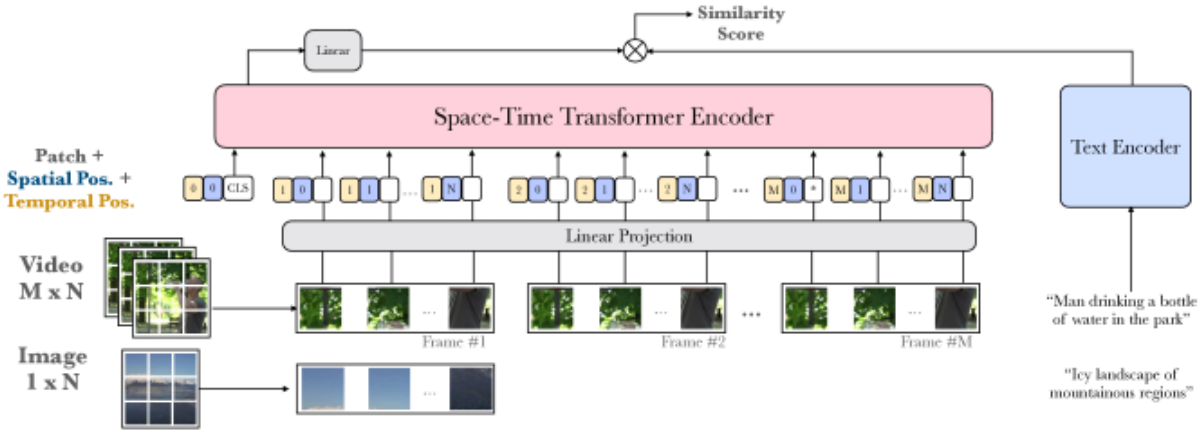
4) Comparing Architecture of CNN-RNN Hybrid vs TimeSformer

A) Using CNN-RNN Hybrid Architecture:

The dominant approach in NLP, which are still very popular to date, is to use a series of bi-directional CNN-RNN networks (RNN) to encode the input sequence and to generate(decode) a sequence of output. The main constraint for a recurrent model is its capacity of handling long sequence. It is the natural way we treat a sentence, from left to right(or in contrast), one word after another. But also due to this character, RNN is restricted in parallelization of computation, because one state is obliged to wait for the computation of its previous state. And it is still constrained to handle super-long sequences. The Seq2Seq model based entirely on CNN can be implemented in parallel, and this reduces a lot of time on training But it occupies a lot of memory, we could also see that it needs many tricks to ensure the last performance, and most importantly, the parameter adjustment on large data volumes is not easy with so many tricks.

B) Video Classification using Transformers: TimeSformer

Transformers is a breakthrough in this domain and It enables the parallelization and improves the performance of attention mechanism by introducing self-attention and also divided attention. Facebook AI has built an entirely new architecture for video understanding. It is the first video architecture that's based purely on Transformers, which in recent years have become the dominant approach for many applications in natural language processing (NLP).



The major differences with respect to the architecture of a CNN-RNN Hybrid to a Space-Time Transformer (TimeSformer) are:

- Adapts the standard Transformer architecture to video by enabling spatio-temporal feature learning directly from a sequence of frame level patches.
- We propose a more efficient architecture for spatiotemporal attention, named “Divided Space-Time Attention” (denoted with T+S), where temporal attention and spatial attention are separately applied one after the other. This architecture is compared to that of Space and Joint Space-Time attention
- There is only one small detail that makes them different from Vision Transformers, you have to take into account not only space but also time. In this case in fact, when we go to calculate the attention and we cannot look at the frames as isolated images but we should find some form of attention that takes into account the variation that occurs between consecutive frames as it is central in the evaluation of a video.

The TimeSformer (from ‘time-space transformer’) divides each video frame into sixteen non-overlapping patches. This example illustrates TimeSformer processing the single patch colored blue, which is part of a three-frame video. The red and green coloring illustrate the idea of attention – the red and green patches are what the analysis of the blue patch ‘pays attention to’.

Paying attention is this case means that the TimeSformer's analysis of the blue patch is based on calculations that combine the characteristics of the blue patch, the red (spatially relevant) patches, and the green (temporally relevant) ones. The analysis of a whole frame combines the characteristics of all the patches in the frame. Although this example shows three frames being used to analyze the blue patch, the operational system actually uses up to 96 frames to analyze every patch in every frame.

5) Model Building: Code and Execution

A) Verifying CNN-RNN Hybrid model's output (Accuracy):

Test Accuracy for the UFC11 Dataset using CNN-RNN Hybrid Architecture was observed. Here, I have taken few samples of data, have run 30 epochs and have observed a test accuracy around 69%. Even the training time for the data was more. This model was computationally not sufficient for me to check the accuracy when I tried to run few videos each with more than 1 minute of length.

```
# Utility for running experiments.
def run_experiment():
    filepath = "/tmp/video_classifier"
    checkpoint = keras.callbacks.ModelCheckpoint(
        filepath, save_weights_only=True, save_best_only=True, verbose=1
    )

    seq_model = get_sequence_model()
    history = seq_model.fit(
        [train_data[0], train_data[1]],
        train_labels,
        validation_split=0.3,
        epochs=EPOCHS,
        callbacks=[checkpoint],
    )

    seq_model.load_weights(filepath)
    _, accuracy = seq_model.evaluate([test_data[0], test_data[1]], test_labels)
    print(f"Test accuracy: {round(accuracy * 100, 2)}%")

    return history, seq_model


_, sequence_model = run_experiment()
```

```
Epoch 00008: val_loss did not improve from 1.47325
Epoch 9/10
13/13 [==============================] - 0s 20ms/step - loss: 0.6740 - accuracy: 0.8627 -
```

```
Epoch 00009: val_loss did not improve from 1.47325
Epoch 10/10
13/13 [==============================] - 0s 20ms/step - loss: 0.6519 - accuracy: 0.8265 -
```

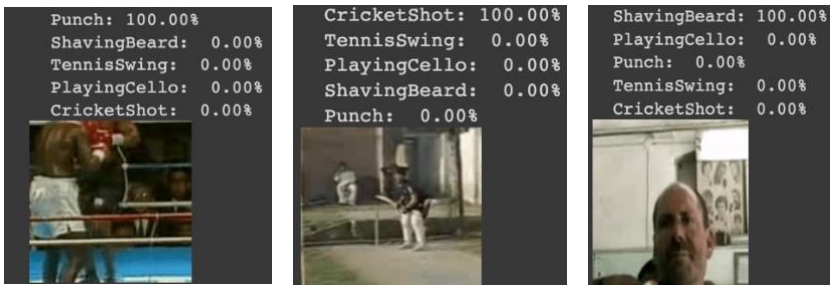
```
Epoch 00010: val_loss did not improve from 1.47325
7/7 [==============================] - 1s 5ms/step - loss: 1.3806 - accuracy: 0.6875
Test accuracy: 68.75%
```

<u>Cons Observed:</u>

- Less Test accuracy
- Takes more than 25minutes just to run few epochs
- More training times
- Wrong action prediction in few cases

B) Verifying the TimeSformer vision transformers output (Accuracy)

Test accuracy which was observed on the same dataset with 12 Epochs was around 87.38%
Test accuracy which was observed on the same dataset with 25 epochs was around 76%
Test accuracy which was observed for 50 epochs was 82.76%.



From this, firstly I have observed that, using Timesformer, I have achieved higher accuracy when compared to the CNN-RNN hybrid architecture.
Secondly, training time was less.
Also, I have got to check the action classification on the few videos with lengths > 1minute i.e 75 seconds, 89 seconds and 94 seconds.

I tried running by taking a sample of data with video lengths >2 minutes approx..
I was not able to run it on my system due to hardware constraints.

```
# main classes

class TimeSformer(nn.Module):
    def __init__(
        self,
        *,
        dim,
        num_frames,
        num_target_frames = 4,
        image_size = 224,
        patch_size = 16,
        channels = 3,
        out_channels = 3,
        depth = 12,
        heads = 8,
        dim_head = 64,
        attn_dropout = 0.,
        ff_dropout = 0.,
    ):
```

```
16/16 [==============================] - ETA: 0s - loss: 2.6743e-04 - accuracy: 1.0000
Epoch 25: val_loss did not improve from 0.48474
16/16 [==============================] - 8s 503ms/step - loss: 2.6743e-04 - accuracy: 1.0000 - val_loss: 3.6321 - val
_accuracy: 0.5222
7/7 [==============================] - 1s 169ms/step - loss: 0.8120 - accuracy: 0.7589
Test accuracy: 75.89%
```

6) Contributions

- By using TimeSformer- A new type of video classification approach which uses both space and time complexity, I was able to accurate video classification on a large dataset.
- Also, I have provided the accuracy comparisons of using a traditional approach to this latest approach.
- Training longer video clips, upto several minutes long which cannot be done efficiently with regular CNN-RNN architectures and Transformers is observed using this method.
- I used 2 different datasets, one with video lengths which are lesser than 30 seconds and other dataset which has videos length more than 1 minute long. Achieved good accuracy in both these datasets.
- Also, the future approach is to train more powerful TimeSformer variants like TimeSformer-HR (operating on 16-frame clips sampled at 448x448 spatial resolution), and TimeSformer-L (operating on 96-frame clips sampled at 224x224 spatial resolution)and we need a GPU with more than 32GB of memory.

7) Conclusion

TimeSformer (from Time-Space Transformer) achieves the best reported numbers on several challenging action recognition benchmarks, Furthermore, compared with modern 3D convolutional neural networks (CNNs), TimeSformer is roughly three times faster to train and requires less than one-tenth the amount of compute for inference. This is an important step toward supporting applications requiring real-time or on-demand processing of video.

- Achieved Higher accuracy, 3-times less training time compared to the generic transformer's architecture.
- Successfully tested on few videos with length greater than 1 Minute.
- TimeSformer operates on **8*224*224** video clips.
- For Videos with larger frames like 16*448*448 and 96*224*224, we can use the variants of TimeSformer like TimeSformer-HR and TimeSformer-L
- Facebook Meta AI currently uses this and they have been getting better outputs compared to their previous transformer Architecture
- This feature is also being implemented by MXPlayer, a streaming platform and currently they are in the testing phase.

8) Appendix

Source Code:

- https://github.com/adasu2/64061_ML.git

References

- http://proceedings.mlr.press/v139/bertasius21a/bertasius21a.pdf
- https://ai.facebook.com/blog/timesformer-a-new-architecture-for-video-understanding/
- https://medium.com/lunit/timesformer-is-space-time-attention-all-you-need-for-video-understanding-5668e84162f4
- https://imerit.net/blog/learning-common-sense-from-video-all-pbm/
- https://www.topbots.com/transformers-timesformers-and-attention/