Spatio-Temporal Attention for Video Action Recognition.

Mogillapalle Venkata Sai Teja , Dhanush Kumar Akunuri

March 9, 2020

1. Subject Presentation

Image classification is a process of classifying the image into one of the predefined classes. One extension to this is video action recognition, which is a process of inferring actions from video clips using attention visualization. Action recognition has many applications in various areas like CCTV surveillance where the actions could possibly be identified without human intervention. Attention maps are a great boost for image classification and video action classification, as they focus on the important regions in the image required for classification ignoring the other regions. This greatly improves the classification accuracy. Extracting attention maps in videos is a challenging task as the videos contain both spatial and temporal information. It is also computationally very expensive to process a video rather than an image as we need to consider 3D data instead of 2-dimensional data. This problem has gained a lot of attention in recent years due to its vast applications. Karpathy et al. 2014 [1] applied CNN to extract features from each frame and then fused all the features to predict the class to which the video clip belongs to. Later, Simmoyan et al. 2014 [2] introduced a two-stream architecture where a network takes Spatial and temporal streams independently. The spatial stream processes the spatial data from the video frames while the other uses optical flow(temporal). At last, both the streams are fused using late fusion. Another work introduced by Du Tran et al. 2014 [3] used 3D convolutional kernels on spatio-temporal cube for finding the attention maps for temporal data.
We have implemented the solution for the action recognition problem as described by Zhenyan et al. [4] where the video is broken down into sequence of images and convLSTM is used to find the temporal relation of the images for action recognition and classification. HMDB-51 data, which contains videos of 51 classes/actions, is used for training the model. It is available in the internet and can be downloaded directly. Each action has around 100 video clips and there are a total of 6766 video clips. We have taken a subset of the data for our experiment.

2. Method

We have implemented the solution as described by [4] for video attention recognition. Initially, any video is partitioned into 30 equally spaced frames which are used as a sequence instead of all the frames in the video. This drastically reduces time taken to train the model. The author describes an encoder, which is used for extracting features blocks from an image. Here we are using a VGG-16 network, which is a pre-trained model with ImageNET data. VGG-16 has a simple network architecture compared to the GoogleNET architecture and shows similar accuracy. The feature cube is extracted from the last convolution layer of the VGG network which has a size of 7x7x512. 7x7 describes the size of a single feature map and there are a total of 512 feature planes.
There is a soft-attention block, which calculates the attention map for an image. Initially, the feature cubes computed from the last conv layer of VGG network are compressed into a vector form(by taking the average of the every 7x7 channel of the feature block) and are added with 'context vector' computed for the previous frame. For obtaining the attention map of the image, the computed summation is sent into softmax layer and then reshaped into the initial dimensions. This results in a probability distribution for the feature maps created by the encoder. The probability value indicates the importance of the pixel in the image. Larger value implies more important region and lower value indicates less important region.
In our implementation, ConvLSTM(Convolutional LSTM) has been used instead of LSTM as the former one preserves the spatial correlations between the pixels which is a crucial thing for attention recognition and video classification. Firstly, the output of the encoder(the last conv output from VGG-16 network), is multiplied(element-wise multiplication) with the attention map generated from the soft-attention block. The dimensions of the results are not altered. This is given as input to the convLSTM block and the output of the LSTM is not only used for predicting the class of the input image, but also used as input for computing the attention map of the next image. In this way, the attention maps are calculated based on the input image features and also from the previous output of the LSTM block.

3. Implementation

We have used HMDB-51 dataset for our problem. This dataset consists of 6766 videos from 51 different classes. This is a very large dataset and considering the model we need to train(having parameters in the range of 100,000), it takes a huge amount of time and resources for training the model. As we have limited resources(RAM and GPU), we thought of considering a subset of the dataset for training. We have taken videos of 10 classes instead of the whole dataset. We have divided the dataset into training, testing and validation sets. Around 730 videos are used for training, 220 videos for validation and 78 videos for testing. Each

video clip is divided into 30 frames and the sequence of these frames represent a video. Very less number of epochs(around 10) are required for convergence. Final output of the model is the predicted class for the image and the probability for that prediction. Attention maps are also obtained for each frame based on the current feature cube and the hidden state output of the previous LSTM block. So, the spatial and temporal information in the video is considered for producing the attention maps and also for predicting the class of the image. Different batch sizes are considered for testing and for the batch size of 15, the model produced a better accuracy.

4. Result

Accuracy of the model is calculated based on the prediction labels for the video data. Prediction is calculated individually for each image frame and compared with the label of the video containing that frame. Our model produced an accuracy of 82% for the test data. But our focus here is more on the attention maps generated for the temporal data rather than the classification results. The attention maps generated from the model are blended with the original image to obtain the resulted image with attention maps. These are shown in fig(1) and fig(2) for the actions, 'Cart Wheel' and 'Brush Hair' respectively.

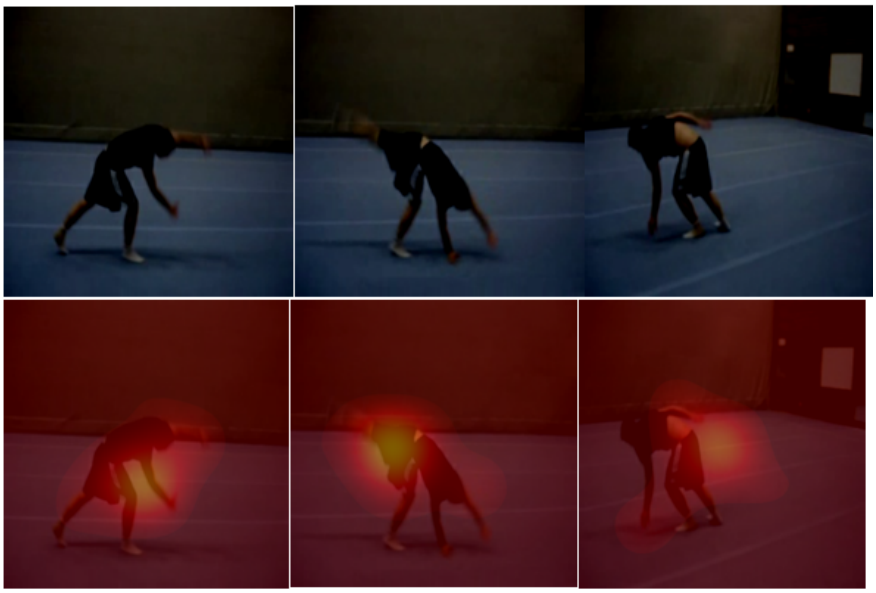

Figure 1: The top 3 images represent the original input frame sequence. The bottom 3 images are the result of overlap of heatmaps with original images. These are the images for action "cart wheel".

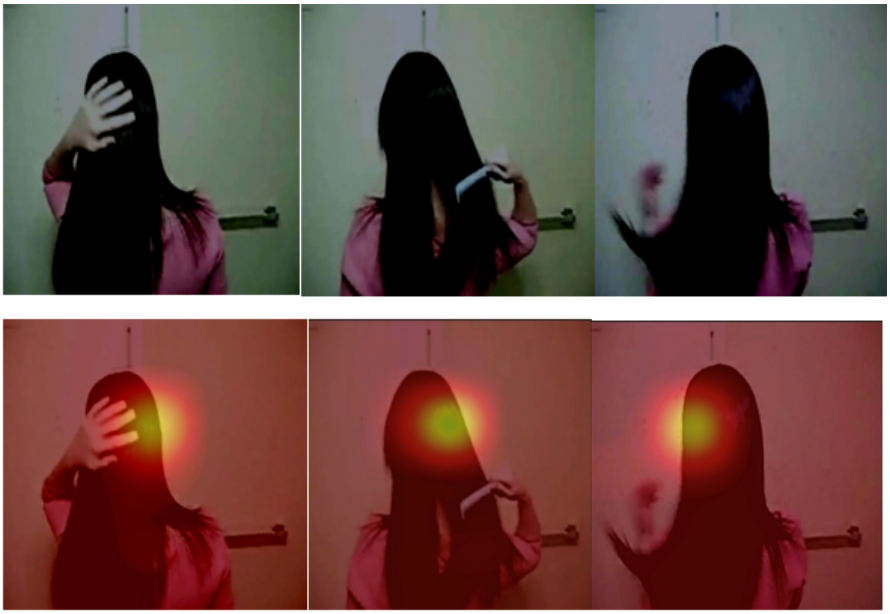

Figure 2: The top 3 images represent the original input frame sequence. The bottom 3 images are the result of overlap of heatmaps with original images. These are the images for action "Brush Hair".

Fig(3) represents the loss function for the training and validation data with respect to the epochs. We can see that the model converges with very few epochs.

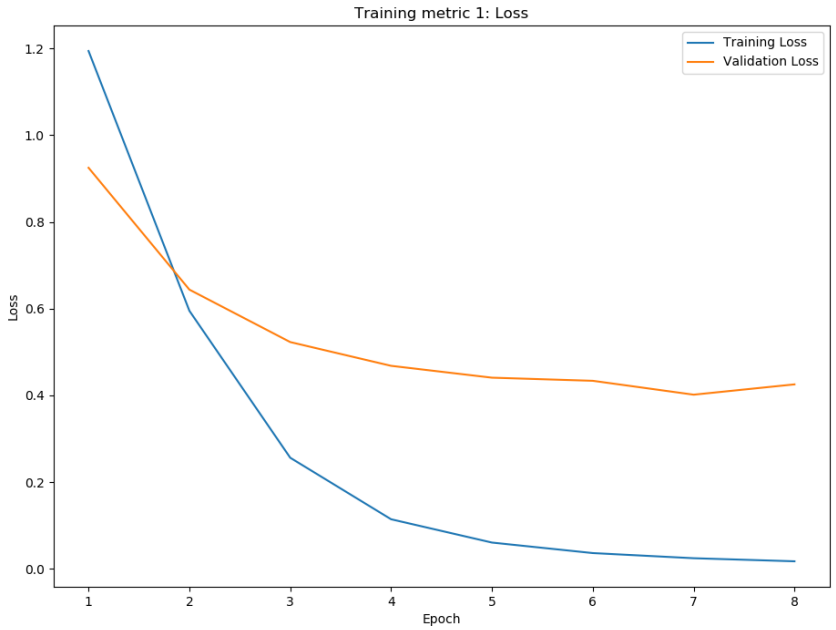

Figure 3: Loss vs epoch plot for Training and Validation data

5. Discussion

As we can see from the attention maps that the model is able to identify the attention areas correctly from the figures (1) and (2). ConvLSTM has worked well in our case to identify the attention maps in the frames. This is definitely an improvement to the performance compared to the model with just LSTM instead of convLSTM. Though there can be no measure like accuracy to figure out how well the attention maps are for the given input, we can clearly identify it by looking at the results. Despite getting high accuracy for the action prediction, the model doesn't correctly identify the attention maps for some cases like when there is more noise in the videos, movement of multiple people/objects in the video, drastic movement of the camera position while recording the video etc. Also, we just trained the model with only 10 classes to reduce the computational time and cost as we had only limited resources for training. In addition to HMDB dataset, training with additional datasets like THUMOS and Kinetics-700 might provide better results for attention maps.We have learnt how to generate attention maps for single image and also for temporal data through this project and how to include the temporal component while training. Our implemented model has somewhat simpler architecture and performs well for the tested data.More complex models might perform even better by training it with more data but there would be a compromise on time, resource and expenses for better performance.

References

[1] Andrej Karpathy, George Toderici., Sanketh Shetty., Thomas Leung., Rahul Sukthankar., and Li Fei-Fei. Large-scale video classification with convolutional neural networks, 2014.

[2] Karen Simonyan. and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos, 2014.

[3] Du Tran., Lubomir Bourdev., Rob Fergus., Lorenzo Torresani., Manohar Paluri., Facebook AI Research., and Dartmouth College. Learning spatiotemporal features with 3d convolutional networks, 2015.

[4] Zhenyang Li., Kirill Gavrilyuk., Efstratios Gavves., Mihir Jain., and Cees G.M.and Snoek. Videolstm convolves, attends and flows for action recognition, 2018.