Action Recognition based on Xception+LSTM and Two-stream Techniques

Mostafa Hegazy
Innopolis University
Innopolis, Russia
m.hegazy@innopolis.university

Mohamed Ahmed
Innopolis University
Innopolis, Russia
o.ahmed@innopolis.university

***Abstract*—The Two-stream convolution neural network (CNN) has proven a great success in action recognition in videos. The main idea is to train the two CNNs to learn spatial and temporal features separately, and two scores are combined to obtain final scores, In this paper, we compare the results of the Two stream CNN against the state-of-the-art CNN+LSTM technique which proved to have a very good performance in classifying actions in videos. Xception model was used as a base model for feature extraction in all the proposed models. We explore the effect of using the Xception model as a base model and the results obtained from both the Two stream model and the Xception+LSTM model.**

I. Introduction

Video-based action recognition is a very challenging task for machine learning and computer vision due to various reasons. Such as the large size of the dataset, the resolution of each frame in the video, the variable length of each video, and the dynamic nature of the video where each frame tells a part of the story. Many approaches have been developed to solve the action recognition problem such as Convolutional Neural Networks (CNNs) along with Long-Short Term Memory Recurrent Neural Networks (LSTMs) to capture the dynamic nature of the videos and Two stream networks that learns both the temporal and spatial features from the videos. In this paper, we explore the two methods above to compare their performance. In the first method, we use the Xception model as a feature extractor along with the Time-Distributed layer to apply the same filters to each frame in the input to produce one output per input. Making it suitable for the LSTM layer which needs the input to have a chronological order to find what is useful and what's not. The action in the video happens in chronological order. The second method, which consists of two neural networks working in parallel to predict the action, the first neural network will learn the spatial features in the frame while the second neural network will learn the temporal features and then fuse the two outputs to produce the final prediction.

II. Related Work

Many of the action recognition methods extract high-dimensional features that can be used within a classifier. These features can be hand-crafted. In the traditional computer vision approach Abdulmunem et. al. [1] used handcrafted features by extracting saliency guided 3D-SIFT-HOOF features. these features are fed to an SVM model after encoding them using the bag of visual words approach. This approach had its drawbacks because of handcrafted features so now the new approaches tend to focus on utilizing unsupervised feature extraction methods and deep learning models. For instance, the method proposed by Simonyan et. al. [2] [6] [7] decomposes video into spatial and temporal components by using RGB and optical flow frames. These components are fed into separate deep ConvNet architectures, to learn spatial as well as temporal information about the appearance and movement of the objects in a scene. Each stream is performing video recognition on its own and for final classification, softmax scores are combined by late fusion. The authors compared several techniques to align the optical flow frames and concluded that simple stacking of L= 10 horizontal and vertical flow fields performs best. In the paper [3], Donahue et al. introduced a long-term recurrent convolutional network that extracts the features from a 2D CNN and passes those through an LSTM network to learn the sequential relationship between those features.

III. Implementation

A. Dataset

UCF11 dataset is used, which has 11 classes and a total of 1600 videos approximately equally distributed, each frame within the video is 240×240 pixels. It contains 11 action categories: basketball shooting, biking/cycling, diving, golf swinging, horseback riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog. This data set is very challenging due to large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc.

Fig. 1. Examples from the UCF11 Dataset

B. Keras video Generator

we had to address the problem of variable video length, So the first step was to extract a fixed number of frames from each video, so we used the Keras-video-generator library which takes the videos, segments each video into parts, and picks one frame from each part, It can also make data augmentation. The number of frames is decided by the support of the hardware. In our case, that number is five.

C. Xception+LSTM

The first approach that we use is the Xception+LSTM approach, We decided to use Xception [4] by google "Extreme version of Inception" model as our base model for feature extraction to test its performance along with a single layer of LSTM and two fully connected dense layers for the final classification of the action from the video.

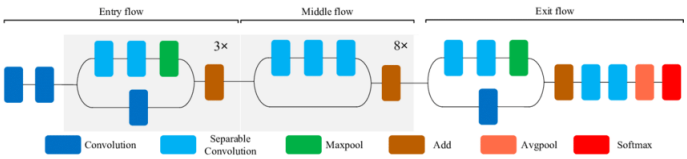

Fig. 2. Xception model block diagram

We fine-tuned our Xception model by freezing all layers except the last four layers to increase the performance of our model while utilizing state of the art techniques such as learning rate scheduling and early stopping.

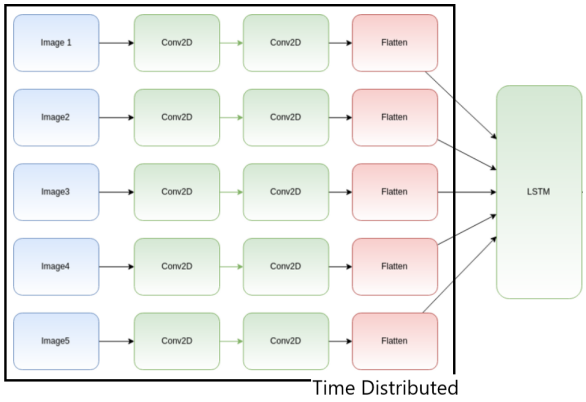

Fig. 3. Time Distributed block diagram

The Time-Distributed layer was used on the Xception base model so that we can apply the feature extraction network on each frame in the video at the same time without learning new weights for each frame. The output of the base model was flattened and transferred to the LSTM layer for dynamic feature extraction and then the output was transferred to the fully connected dense layers for the final classification.

D. Training parameters

The batch size was chose as five. The model is trained using categorical cross-entropy losses on the outputs at all time steps. Input video frames are sub-sampled by using Keras-video generator which takes 5 frames from the video.

E. Two Stream

Video is a collection of spatial and temporal information. Information is static image appearance in a spatial stream; it only depicts scenes and objects. In the temporal stream, information is the movement of objects between consecutive frames, conveys the orientation of the camera and objects.

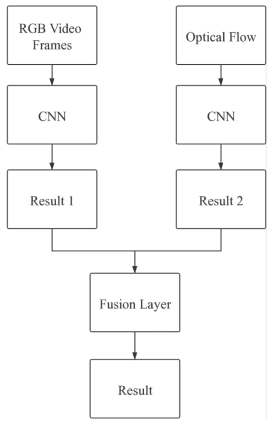

Fig. 4. Two stream block diagram

For spatial information, RGB frames are used. For temporal information, dense optical flow frames are used to extract the motion of objects across the video. Each of these streams is processed identical and independent deep convolutional neural network models. Specifically, the RGB image is fed to the spatial stream CNN. For temporal stream, the stack of optical flow images is fed as input.

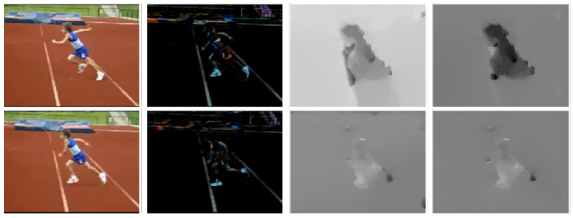

Fig. 5. Optical flow example

Optical flow images are a combination of horizontal and vertical convoluted images. The number of optical flow images(L) is set to four. Because the optical flow images consist of both horizontal and vertical convoluted images, the total number of flow images is set to $2L = 8$. Finally, spatial and temporal streams are individually trained end-to-end, and the output of two streams are combined to get the final classification decision.

F. Spatial Network

Xception model was used as a feature extractor along with an LSTM layer and three dense layers at the end for the final classification. The input to the network is RGB frames

whose size is 240x240 and sampled at a rate of five frames per second. This network is supposed to capture the spatial information from the video.

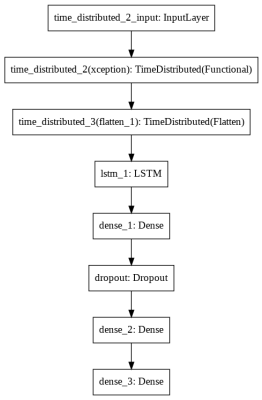

Fig. 6. Spatial Network

G. Temporal Network

Stacking 10 optical flow images for the temporal stream has been considered as a standard for two-stream ConvNets. We don't follow the standard instead we are using 8 optical flow images. In particular, using a pre-trained network and fine-tuning has been confirmed to be extremely helpful despite differences in the data distributions between RGB and optical flow. So we use the same base model from the spatial network and we change only the classifier network.

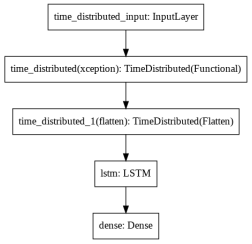

Fig. 7. Temporal Network

H. Training parameters for Two stream Network

The batch size was chosen as four. The model is trained using categorical cross-entropy losses on the outputs at all time steps. Input frames are the concatenation of both the x and y channels from the optical flow frames.

IV. Experiments and Evaluation

The training data was split into train and validation data during the training phase. A private dataset was kept to test the models after training. The metric used for testing and evaluation is Accuracy. The test dataset contained 404 videos approximately equally distributed between the 11 classes. The number of epochs used in the training is 30. RTX2060 is the GPU used to train the models.

TABLE I
Model Performance

Network Type	Accuracy		
	Training	***Validation***	***Testing***
Xception+LSTM	98.9%	99.1%	89.2%
Spatial Network	98.9%	99.1%	89.2%
Temporal Network	79.85%	66.67%	64.58%
Two Stream	–	–	90.05%

V. Analysis and Observations

The training time of the Spatial network with significantly lower than the temporal network because of the reduced number of frames extracted from each video. The accuracy of the spatial network was found to be sufficient for the general classifying problem with good accuracy. Two Stream network is found to be very powerful if the optical flow data is available. The accuracy increased from 89% to 90% by fusing the output of the temporal and spatial networks at the softmax layer.

TABLE II
Parameters

Network	Training time (min)
Xception+LSTM	36.03
Spatial Network	36.03
Temporal Network	95

VI. GitHub link

click here

VII. Conclusion

We proposed the use of the Xception model as a base model for feature extraction in both the Spatial and Temporal networks which proved to have good performance in extracting the necessary features required for the classification task of the action recognition. Further testing and research are needed in the action recognition field. In the future, we would like to test the performance of the Xception model on more complex and bigger datasets such as the UCF101 dataset.

References

[1] A. Abdulmunem, Y.-K. Lai, and X. Sun. Saliency guided local and global descriptors for effective action recognition. Computational Visual Media, 2(1):97–106, 2016.
[2] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. CoRR, abs/1406.2199, 2014.
[3] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In CVPR, 2015
[4] Xception: Deep Learning with Depthwise Separable Convolutions
[5] J. Liu, J. Luo and M. Shah, Recognizing realistic actions from videos "in the wild", CVPR 2009, Miami, FL.
[6] Cheng Dai, Xingang Liu, Jinfeng Lai, "Human action recognition using two-stream attention based LSTM networks".
[7] Harshala Gammulle, Simon Denman, Sridha Sridharan, Clinton Fookes, "Two Stream LSTM: A Deep Fusion Framework for Human Action Recognition Image and Video Laboratory"