Classification of Events in Indian Movie Scenes

Mounika Chadalavada (S20160020177), Parkhi Mohan (S20160010061), Sree Pragna Vinnakoti (S20160010106)

{mounika.c16, parkhi.m16, sreepragna.v16}@iiits.in

Indian Institute of Information Technology, Sri City

***Abstract*—This paper presents a methodical approach to classify events in Indian movie scenes. The nine categories into which the scenes shall be categorised under are - Chases, Dance, Eating, Fight, Heated Discussions, Normal Chatting, Romance, Running and Tragedy. To achieve this we have generated spatial and temporal features of the input data and given to a training model. Once the training and cross validation has been performed, testing is done and the scene category is given as output.**

***Keywords*—Scene detection, VGG16 model, Optical Flow, Tensorflow, Keras, LSTM, RNN, Transfer learning**

I. INTRODUCTION

Scene classification involves identification of various events from the movie scenes in which an event may or may not have occurred throughout the entire video. It mainly involves capturing spatio-temporal context across frames. This is a natural extension of image classification to multiple frames and then aggregating the predictions from each frame. To serve this purpose one can use deep learning techniques to get decent results.

Before deep learning, traditional CV algorithms were applied where local high-dimensional visual features that describe a region of the video are extracted either densely or at a sparse set of interest points. The extracted features get combined into a fixed-sized video level description. One popular variant to the step is to make bag of visual words for encoding features at video-level. A classifier like SVM is trained on visual words for final prediction. SVM gives best results when the dataset is small but fails to give accurate results in noisy environment and this is when deep learning is preferred.

Training a deep network takes weeks to train using many machines equipped with expensive GPUs. As our dataset is very small the network does not result in that high an accuracy. This is where transfer learning plays a key role. In transfer learning, the models are pre-trained on a large scale image classification problems. A pre-trained model is trained on a different task than the task at hand but provides a very useful starting point because the features learned while training on the old task are useful for the new task. The pre-trained model used is this paper is VGG16 [2, 4].

In this paper we have proposed a method that will classify events in Indian movie scenes into nine different categories - Chases, Dance, Eating, Fight, Heated discussions, Normal chatting, Romance, Running and Tragedy.

A. VGG16 Model

VGG16 is a deep convolutional neural network with weights pre-trained on ImageNet database. The ImageNet database, built upon the hierarchical structure of Worldnet, contains more than 3.2 million cleanly annotated images with weights. Since the pre-trained VGG-16 model has learned to extract features from images that can distinguish one imageclass from another, they have shown to achieve excellent performance even when applied to image recognition and classification datasets in other domains. This neural network has sixteen convolutional layers, five max-pooling layers, followed by a fully-connected layer, with the final layer as the soft-max layer. Rectification nonlinearity (ReLu) activation is applied to all the hidden layers. The model also uses dropout regularization in the fully-connected layer [5].

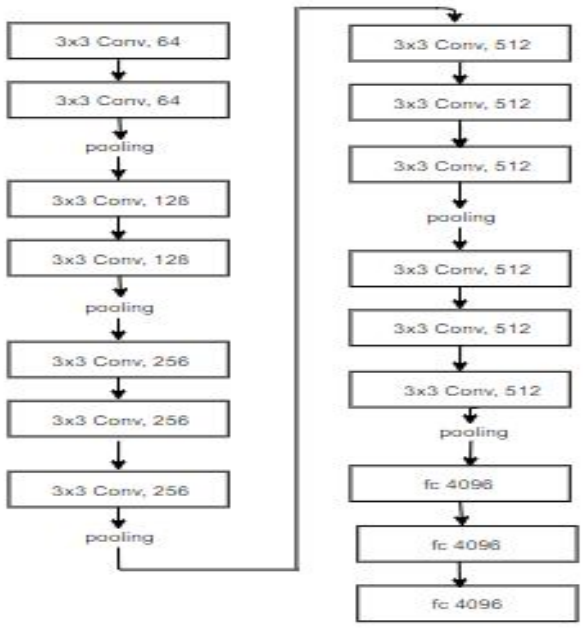

Fig. 1. Architecture of VGG16 model

B. Soft-max function

Softmax function calculates the probability distribution of an event over 'n' different events. This function will calculate the probabilities of each target class over all possible target classes. Later the calculated probabilities will be helpful for determining the target class for the given inputs. The output range of this function is 0 to 1.

Softmax function :

$$\sigma(x_j) = \frac{e^{x_j}}{\sum_i e^{x_i}}$$

C. Optical flow

Optical flow [3] is the pattern of apparent motion of objects, surfaces and edges in a visual scene caused by the relative motion between the observer (an eye or a camera) and the scene. Optical flow is mainly used to track the motion of an object or blob across frames in a moving sequence which is very helpful in scene classification.

D. LSTM

Long Short Term Memory networks, usually just called "LSTMs" are a special kind of RNN, capable of learning long-term dependencies. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. LSTM networks manage to keep contextual information of inputs by integrating a loop that allows information to flow from one step to the next. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. LSTM networks are well-suited for classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. The output of the LSTM network is modulated by the state of contextual state cells. LSTM predictions are always conditioned by the past experience of the network's inputs. LSTMs help preserve the error that can be back-propagated through time and layers. By maintaining a more constant error, they allow recurrent nets to continue to learn over many time steps (over 1000), thereby opening a channel to link causes and effects remotely.

II. DATASET

The given dataset contains 227 Indian movie scenes divided into nine different categories - Chases, Dance, Eating, Fight, Heated discussions, Normal chatting, Romance, Running and Tragedy. Each class contains 9 to 77 video clips.

TABLE I. DATASET STATISTICS

S.No.	Class label	No. of videos in dataset	Average length of the videos
1.	Chases	23	2 min 30 sec
2.	Dance	77	30 sec
3.	Eating	9	50 sec
4.	Fight	20	3 min
5.	Heated discussions	21	2 min 30 sec
6.	Normal chatting	22	1 min 15 sec
7.	Romance	20	1 min 10 sec
8.	Running	15	55 sec
9.	Tragedy	20	2 min 30 sec

The length of the videos ranges from a minimum of ten seconds to a maximum of four minutes three seconds. All the videos are in the .mp4 format.

Fig. 2. Snapshots of dataset from each class.

III. PROCEDURE

A. Pre-Processing

Before generating the features from the given dataset, the dataset needs to be pre-processed to obtain better results. The dataset contains few videos where the main event occurs only for a smaller duration in the video, such videos have been trimmed accordingly. Next we converted the .mp4 format files into .avi format files for better

processing. Later scaling has been done on all the input data videos to resize them to a size of 224x224 using ffmpeg library implemented using python.

B. Generation of Optical Flow

For generating optical flow from the videos we have divided the videos into frames and used the previous and current frame to generate an optical flow image. This is implemented using opencv library in python. All such generated images are written into a video file and an optical flow video of the input video is generated. This procedure is applied to every video of the dataset and optical flow videos have been obtained.

Fig. 3. Snapshots of optical flow generated from the dataset

C. Generating features

Features required to train the model are generated using the pre-processed data. For feature generation we have divided each video into frames with time step of one second. Later we convert the list of frames into a numpy array and send the frames into the VGG16 pretrained model using keras library to generate output predictions for the input samples. The model returns the predictions in the form of a numpy array. This procedure is applied on every video of an individual class and all the such predictions are combined and written into a single feature file. Therefore by the end of this procedure we will have nine different feature files depicting features of their individual classes. The same procedure is applied on the optical flow videos generated earlier. Another nine feature file will be generated by the end of procedure.

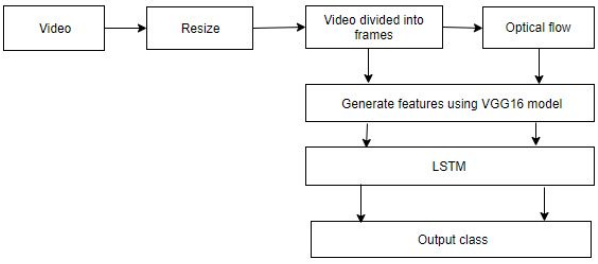

Fig. 4. Flow chart of the procedure followed

D. Training and Cross Validation

Once the feature files are generated, the model is trained using the generated features. The features are loaded as training data and labelled. Once the labelling is done, we initialize and configure (optimizer, loss and metrics) the model. We have popped last two layers of the pre-trained VGG16 model and have added few lstm layers (using keras and tensorflow) and constructed a fully connected layer and an output layer. Once the configuration is done, we start our training. Along with the training, cross-validation is also performed to further finetune the parameters to get the best model and check for overfitting. We have trained the data for one hundred epochs. Once the training is done the model is saved in a file.

E. Testing Model

Once the training is done, the model that has been saved is used for testing. As a part a of testing we give a video file as input. The input video is divided into frames and each frame is used for making predictions. Each and every frame is assigned a class name and score. The class with maximum prediction score is given as output category.

IV. RESULTS

A scene from an Indian movie is given as input to the trained model, the model was successfully able to predict the category of the scene.

While training the model we have calculated the accuracy and loss (sparse categorical cross entropy) of the model. We observed that for each epoch value there has been a gradual increase in accuracy of predictions and a gradual decrease in loss. We have plotted the accuracy and loss of the model against the epoch value.

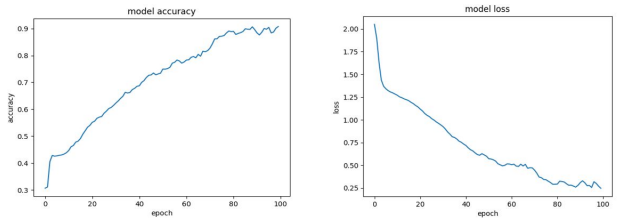

Fig. 5. Accuracy and Loss plotted against epoch (without optical flow)

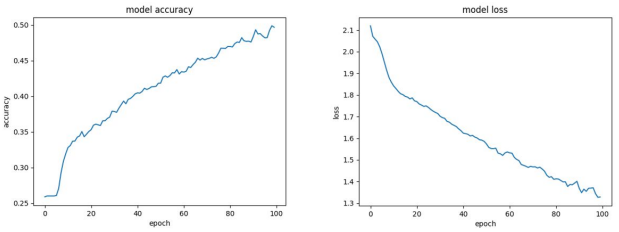

Fig. 6. Accuracy and loss plotted against epoch (using optical flow)

V. SUMMARY AND FURTHER STEPS

With the Indian movie scene database containing very few videos, it is apt to use a pre-trained model for classification for optimum results. We have implemented two techniques for the classification of our dataset - Optical Flow and LSTM.

The results can be further improved by having a proper dataset containing equal number of training videos per category of similar average length.

REFERENCES

[1] Feichtenhofer, Christoph, Axel Pinz, and Andrew Zisserman. "Convolutional two-stream network fusion for video action recognition." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[2] *Canziani, Alfredo, Adam Paszke, and Eugenio Culurciello. "An analysis of deep neural network models for practical applications." arXiv preprint arXiv:1605.07678 (2016).*

[3] *Barron, John L., David J. Fleet, and Steven S. Beauchemin. "Performance of optical flow techniques." International journal of computer vision 12.1 (1994): 43-7*

[4] Gopalakrishnan, Kasthurirangan, et al. Deep Convolutional Neural Networks with Transfer Learning for Computer Vision-Based Data-Driven Pavement Distress Detection. Kasthurirangan Gopalakrishnan, Sept. 2017, www.elsevier.com/locate/conbuildmat. 157(2017)322-330

[5] Simonyan, Karen, and Andrew Zisserman. Very Deep Convolutional Networks For Large-Scale Image Recognition. Karen Simonyan, 10 Apr. 2015, www.robots.ox.ac.uk/~vgg/research/very_deep/. Published as a conference paper at ICLR 2015