MovieScope: Movie trailer classification using Deep Neural Networks

Sivaraman K S & Gautam Somappa
Dept of Computer Science
University of Virginia
{ks6cq, gs9ed}@virginia.edu

Abstract

This paper deals with identifying the genre of a movie by analyzing just the visual features of its trailer. This task seems to be very trivial for a human; our endeavor is to create a vision system that can do the same, accurately. We discuss the approaches we take and our experimental observations.

The contributions of this work are : (1) we propose a neural network (based on VGG) that can classify movie trailers based on their genres; (2) we release a curated dataset, called YouTube-Trailer Dataset, which has over 800 movie trailers spanning over 4 genres. We achieve an accuracy of 80.1% with the spatial features, and 85% with using LSTM and set these results as the benchmark for this dataset. We have made the source code publicly available.[1]

1. Introduction

Deep Learning has propelled advances in the field of Computer Vision. The effectiveness of deep learning in tasks like Object Detection and Recognition lies in the capability to learn rich features from large amount of raw data. The performance of deep learning models in such tasks have improved year-on-year. Similar approaches have also been extended for videos.

1.1. Video Classification

The vast amount of video content that is available has prompted researchers to apply deep learning models and techniques to this domain as well. Interestingly, these models have been able to learn and represent visual features extracted from the video frames. Image-based video classification is a technique where these frame-level features obtained from the fully-connected layers of a deep model are stacked and averaged into video-level representation which is further passed to classifiers for recognition.

This method of stacking deep features across the frames is highly intuitive, but has not performed as accurately; probably owing to lack of a deep enough model that can understand the complexity of the spatial and temporal aspects of a video. Moreover, training CNNs with 3D volumes is time consuming.

Most of the video classification task can be seen for activity recognition. These are shorter (less than 30 seconds) and less dynamic videos, typically comprising of a single activity. We will discuss shortly in detail, some related work in this aspect.

1.2. Movie Trailers

Nowadays, Hollywood makes about 760 movies a year, this is roughly 2 movies every day. That being said, every movie has at least 3 trailers before the final movie rolls out in theatres. Application of Video Classification to movie trailers to predict the genre of the movie is not just an interesting computer vision problem to solve, but also has extended benefits, such as automatically tagging the genre on content-hosting websites like YouTube, Netflix or IMDB, without manual monitoring. Other applications including sentiment analysis for videos uploaded on news-content websites.

Conversely, movie trailers can be considered as a controlled summary of a long movie. If treated as an output, movie trailers can be used to study the subject of Video Summarization, which can potentially lead to other areas of research in computer vision as well.

1.3. Related Work

Video classification has been well researched. The performance however, might not be as glorified as that of image classification. There have been several efforts toward video content understanding, most of them however use hand-crafted features. We discuss some related work happening in the field of video classification using deep neural networks.

Ji et.al, introduced 3D CNN model that operates on stacked video frames, extending the traditional 2D CNN

[1] https://github.com/maximus009/MovieScope

designed for images [4]. Karpathy et al., compared several similar architectures on a large scale video dataset [5]. One of the foremost work was done by Simonyan and Zisserman [9], who introduced the Two-Stream Convolutional Network, which accounts for a fusion of the spatial stream and the temporal stream of a video. Feichtenhofer et al. proposed a better fusion approach to the Two-Stream CNN [3]. These models pertain to activity recognition. The temporal stream is achieved by using Optical Flow Features, which remain one of the most innate features to represent video and motion, upon which a CNN is trained. Ballas et al. have worked on learning video representations using deep CNNs [1].

A more recent approach as proposed by Ng et al. [7] uses LSTMs, which capture the temporal stream by storing the previous sequence features whilst computing the current sequence. propose a deep network for video classification in their paper. They incorporate the use of LSTMs not only on the spatial features, but also (optionally) on the optical flow features, and have reported to improve the benchmark accuracy on established datasets like UCF-101[11] and HMDB-51 [6].

1.4. Related Data Sets

UCF101 is an action recognition data set. It has 13320 videos from 101 action categories. It gives a lot of diversity in terms of actions and with the presence of large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc.

HMDB51 data set contains 51 distinct action categories each containing at least 101 clips. It had around 7000 video clips from a wide range of sources. It is able to fully capture the complexity of video clips found in movies and online videos.

Both these datasets enable recognition of activities over shorter videos, which are not temporally dynamic as a trailer.

2. YouTube Trailer Dataset

2.1. Database schema

Here we describe our dataset for movie trailers on YouTube, which we call the YouTube Trailer Dataset. This dataset was created with the purpose of classifying movie trailers based on genres solely using visual cues. This dataset is the first to provide over 800 Hollywood movie trailers obtained from YouTube, for over 4 genres.

We provide the following metadata for the videos: ID, Movie Name, Genre, Format. These are indexed in MongoDB so they can be queried based on categories and user preferences later on.

2.2. Dataset Download

The trailers are downloaded from playlists on YouTube. We took playlists from verified channels on YouTube like Movieclips Trailers, JoBlo Movie Trailers and manually separated the videos based on genres. The categories we are considering for this work are Action, Romance, Horror and Drama. We realized that some trailers are repeated in multiple genres. However, we are considering each video to belong to a single genre, so the movies were retained in the genre they are most closely related to. We downloaded about 100 videos for each genre for training and 40 videos for testing. The resultant dataset was carefully curated to ensure uniqueness of videos. The dataset is available here.[2]

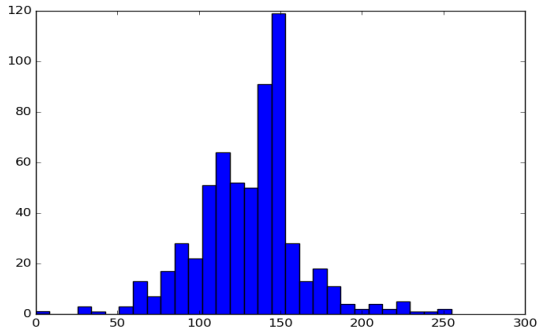


Figure 1. Histogram of Duration of Trailers in ms

2.3. MongoDB Indexing

Once the video files have been downloaded, we indexed it in MongoDB. MongoDB is a noSQL database which uses JSON-like schema. The video metadata is stored in MongoDB so that it can be easily queried based on the developer's requirement.

3. Proposed methodology

Most video classification techniques sample frames uniformly and aggregate the frame features. Since we are dealing with movie trailers, more specifically, the Hollywood movie trailers. The first five seconds or so only display the certification and production house and other information unrelated to what genre the movie might belong to. Same goes for the last few seconds. They only contain the movie banner, the production and cast/crew details. Moreover, trailers have different lengths, but typically, their duration averages between 140 seconds to 160 seconds.

We sample frames at a rate of 1 second, between the 5th second and the 120th second mark, inclusive. This gives us a total of 116 frames per video. Each of these frames are

[2]https://virginia.app.box.com/s/hhe7xeq96d99yuqn1nr1q9yxdhs7ey9r

then passed to the proposed deep model and the resultant features are used to train the classifier.

Figure 2 illustrates the sampling of frames.

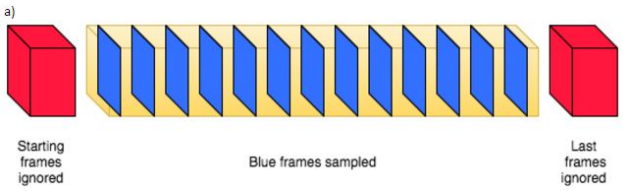


Figure 2. Frame Rate Sampling

3.1. VGG features

VGGNet [10] was introduced for image classification and secured second places for the tasks of image localization and classification on the ImageNet ILSVRC-2014[8]; with a top-5 classification error rate of 7.5% on the validation set. There have been two versions of the network, one a 16-layer network, and a 19-layer. In our model, we use the VGG16 framework. The image is passed to this network and the resultant feature obtained from a single image is a 4096 dimensional vector.

3.2. Loose Labeling of Frames

Intuitively, each video is associated with a genre/label. But we take a slightly different approach. We incorporate the concept of "Loose Labeling" of the frames. Instead of representing the video (which is a collection of frames), we are representing each frame and associating it with the label same as the genre of the video it belongs to. So, instead of averaging the frames to a single representation, we are distributing the same label across all the frames. This can be treated as an uncommon way of augmenting data, only that the labels are augmented and the not the features within.

Let us consider the scenario, that a trailer belonging to romance has some action scenes in the trailer. We are forcing those frames (corresponding to action) to be labeled as romance, simply because they belong to a movie trailer labeled so. This might come across as an attempt at creating a poorly labeled dataset. But our intuition is, that considering the entire dataset, the model will not overfit and thus, learn to generalize when it looks at all the other movie trailers from different genres. We shall discuss this part again, in the Results section.

3.3. Optical Flow

We also implemented a dense optical flow model in this paper. Videos can be decomposed into either spatial or temporal components. The spatial part is already explained above by sampling frames and training our model on them. In this section, we discuss about a ConvNet model that uses Dense Optical Flow. Optical flow is the pattern of apparent motion of image objects between two consecutive frames caused by the movement of object or camera. It is 2D vector field where each vector is a displacement vector showing the movement of points from first frame to second. There are several variations of optical flow based input like:

3.3.1 Sparse Optical Flow

This takes into account good features from consecutive frames and tracks how much the pixel has moved from the one frame to another. Sequences of ordered images allow the estimation of motion as either instantaneous image velocities or discrete image displacements. There is an alternative motion representation called trajectory stacking inspired by trajectory based stacking. trajectory stacking is different from sparse optical flow in a way that it samples at the same location across every frame with the flow sampled along motion trajectories.

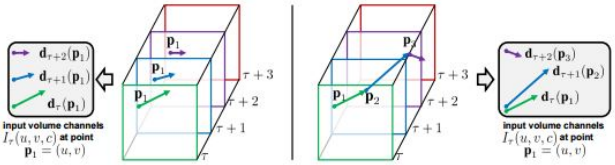


Figure 3. Left- optical flow stacking: samples the displacement vectors d at the same location in multiple frames. Right- trajectory stacking: samples the vectors along the trajectory. The frames and the corresponding displacement vectors are shown with the same colour [9]

3.3.2 Dense Optical Flow

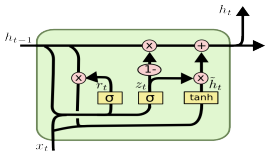
Dense Optical Flow computes the optical flow for all the points in the frame. It is based on Farneback's algorithm [2]. Dense optical flow attempts to give you the flow all over the image - up to a flow vector per pixel, as shown in figure 4. In this paper, we have used dense optical flow for our model. The reason being, this is because the trailers are very dynamic and fast paced. The scenes cut abruptly, so in order to capture each scene as a feature, we realized sparse optical flow will not work on these complex videos. We did try splitting the trailer to multiple shots, and considering each scene for training, just as we used each frame, in the earlier case. But this did not give good results; we are hence, not reporting the results.

3.4. LSTM Model

Long Short-Term Memory (LSTM) cells are primarily used for sequence classification. These memory cells store, modify and access internal states to allow better learning of temporal relatioships in a given sequence. Intuitively, videos can be treated as a sequence of images. Using

Figure 4. Dense Optical Flow

LSTMs for Video classification has already been proposed in [7]. Although, our approach is slightly different, the intent is the same. Unlike their approach, we are splitting the video into 9 frames at a time, and passing the these sequences of fixed number of frames to the LSTM model and stack these respective features to represent the video.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figure 5. LSTM cell

We have used Stacked LSTM model; and have considered a time step of 9, since we are analyzing 9 frames at a time. Like the spatial approach, the input to the LSTM will be the VGG features. The model architecture is shown in the figure 6.

4. Implementation details

4.1. Network Architecture

We incorporate the VGG16 network, and take the output of the FC layer, which results in a 4096 dimensional vector. The VGG network contained the top layers and was pretrained on ImageNet weights and contains 138,357,544 network parameters. For the spatial model, this VGG network is entailed by network designed as shown in figure 6, which contains 8,931,908 network parameters. For the temporal model, the LSTM network is also shown in the figure 6. All fully-connected layers follow ReLU activation, except the last layer which is Softmax output of the number of desired classes, which is 4 in our case.

4.2. Training

We have taken 70 videos for Action, 95 videos for Drama, 99 videos for Horror and 80 videos for Romance. During the training procedure, all the frame features and

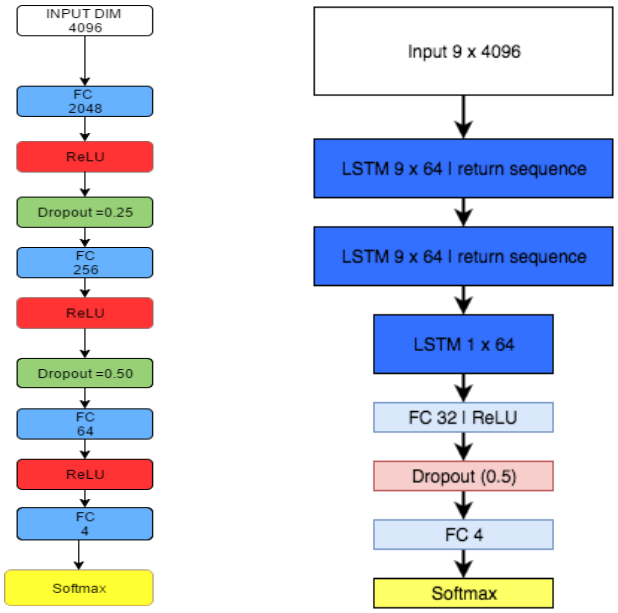


Figure 6. Spatial and LSTM Model

corresponding labels were passed to the model. In all, we gathered around 36,000 vectors; which by the current standards is a small training set. The frames are resized to 224 x 224 from their original size. We used Stochastic Gradient Descent Optimizer at a learning rate of 0.01, without decay, with a batch size of 32 samples. We do not perform an end-to-end training, i.e; only the network defined by us was trained and the VGG model was frozen.

The training was done on the standard AWS g2 instance (NVIDIA GRID K520, 4GB). It took about 10 seconds per epoch. As mentioned earlier, the quick training is mainly due to small training size.

4.3. Testing

The end goal of our project is to test the classification of a video using the trained model. The test set consists 44 trailers from action, 39 from drama, 59 from horror and 39 from romance. Given a video, frames are extracted in a similar way as was done during training; i.e, sampled at every second between the 5th and the 120th second (to preserve consistency). For the spatial model, predictions are made for each frame. The corresponding softmax scores were saved along-with the predicted class. Finally, the class that was predicted the most, among the 116 frames is decided as the genre/class to which the trailer belongs. Similarly, for the LSTM model, frames were sampled at every second between 5th and 120th second, but the model looks at every 9 frames to make the prediction. In total, we would obtain 116/9, which is 13 frame sequences. Predictions were made for these 13 sequences, and whichever class was predicted the most will be decided as the genre of

the trailer video. The scores are then averaged across all the frames/sequences for reference.

5. Evaluation

As mentioned before, our testing set comprises of 44 movie trailers from action, 39 from drama, 59 from horror and 39 from romance genres. The following are the evaluations conducted by us.

5.1. Baseline

As a baseline feature, we extracted the color histograms for each frame and trained the set with a Random Forest Classifier. The feature was a 768 (256 x 3) dimensional vector. The accuracy of this model is close to 60%. This was done to show the effectiveness of features learned using a deep network.

5.2. Evaluation on Test Set

The following are the confusion matrices for the classification tasks using spatial model, optical flow model, and the LSTM model.

Confusion Matrix for Random Forest

Genres	Action	Drama	Horror	Romance	Recall
Action	32	5	1	3	78.0
Drama	6	20	9	12	42.5
Horror	6	3	42	6	73.6
Romance	0	11	7	18	50.0
Precision	72.7	51.2	71.1	46.1	61.8

All values are given in (%) and the accuracy is found to be 61.87

Confusion Matrix for Spatial

Genres	Action	Drama	Horror	Romance	Recall
Action	42	2	2	0	91.3
Drama	2	25	5	9	60.9
Horror	0	3	50	2	90.9
Romance	0	9	2	28	71.7
Precision	95.4	64.1	84.7	71.7	80.1

All values are given in (%) and the accuracy is found to be 80.11

Confusion Matrix for LSTM

Genres	Action	Drama	Horror	Romance	Recall
Action	40	0	0	1	97.4
Drama	3	33	8	1	73.3
Horror	0	1	50	5	89.2
Romance	1	5	1	32	82.0
Precision	90.9	84.6	84.7	82.0	85.6

All values are given in (%) and the accuracy is found to be 85.63.

6. Results

In this section, we display some sample results as given by our model for individual frames from a given trailer.

Figure 7. Frames and its classifications

Some misclassified image examples :

Figure 8. Misclassified images

The sample images above highlight some of the results as predicted by our model. It can be seen that the system has learnt well from visual cues. Our system generalizes well and can predict the genre with high accuracy. To see how well the model has generalized, we test the model with the training data; it was observed that the per-frame/per-sequence output was correct, even though it was trained incorrectly, or loosely, as we had mentioned earlier. This proves that our model has not overfit, and has generalized by looking at the entire training data.

One fallacy that we discovered - very few times, two consecutive frames with minimal or no change in the visual

content might be classified in different genres. Please refer to figure 9.

Figure 9. Similar frames classified differently

We analyzed the Softmax output of such pairs of frames, and the values are marginally close for the two genres that were predicted.

Clearly humans would consider both the frames as the same. But the model made a mistake in this case and misclassified frames. No system is perfect, and sometimes results can be (hilariously) incorrect. More results have shown below:

6.1. Drama

Figure 10. Examples from Drama

The above images were classified as drama by the model. From the frames, the model learns to classify frames with multiple faces/persons as drama.

Figure 11. Examples from Romance

6.2. Romance

The image in figure 11 were classified as romance by the model. Striking features about these frames are that, they are bright, colourful and the frames usually has more than two faces.

6.3. Action

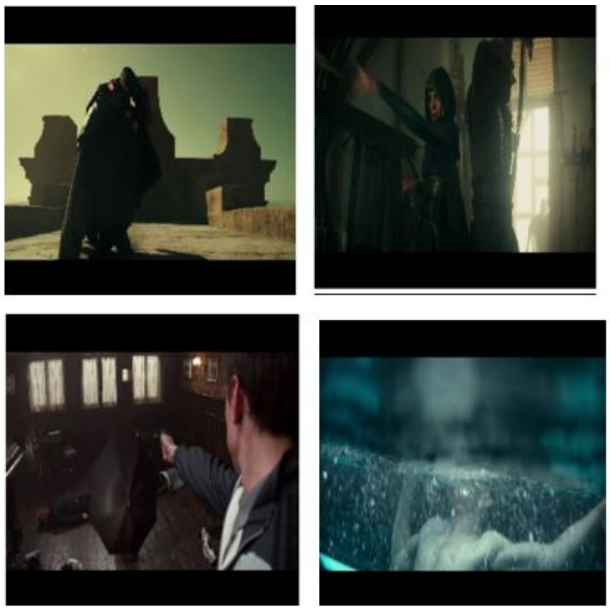


Figure 12. Examples from Action

The images were classified as action by the model. From the frames, we can see that the it denotes a person performing some kind of action and the frame is mostly blurry. The system had learned these from the training videos.

6.4. Horror

The above images were classified as horror by the model. From the frames, we can see that its usually dark and

shadowy and some frames have text.There are some misclassifications though, for example, the last frame which has the words “A Nation drained of natural resources” is classified as horror whereas in reality it is just text on a black screen. Since the system takes into account only spatial features and the frame is mostly contains black color, it is misclassified.

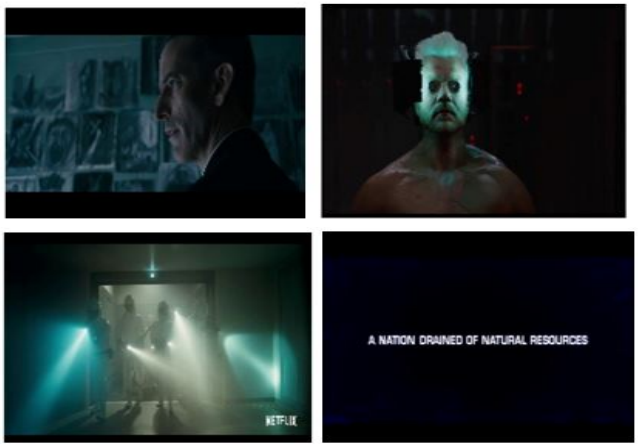


Figure 13. Examples from Horror

7. Conclusion

This paper proposes a novel method to classify videos, specifically movie trailers into four genres namely, action, drama, horror and romance. It is reported that LSTM performs better than using just the single frame features. This shows that LSTMs are able to capture the temporal complexity and can be well exploited to perform tasks such as video classification or video understanding. We have also presented a detailed analysis about the results.

In addition to predicting genres, we have also created a data set that contains movie trailers from the respective genres. We hope to establish the proposed dataset YouTube Trailer dataset as means to research deeper into temporally dynamic videos that will enable in augmenting the work done in video understanding.

8. Acknowledgements

This work was done as a part of the course CS 6501-003: Computational Visual Recognition. Special thanks to our instructor Vicente Ordonez, whose constant guidance and motivation played a huge, encouraging role in making this project possible.

References

[1] N. Ballas, L. Yao, C. Pal, and A. C. Courville. Delving deeper into convolutional networks for learning video representations. *CoRR*, abs/1511.06432, 2015.

[2] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Proceedings of the 13th Scandinavian Conference on Image Analysis*, SCIA'03, pages 363–370, Berlin, Heidelberg, 2003. Springer-Verlag.

[3] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. *CoRR*, abs/1604.06573, 2016.

[4] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):221–231, Jan. 2013.

[5] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. Li. Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 1725–1732, 2014.

[6] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[7] J. Y. Ng, M. J. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. *CoRR*, abs/1503.08909, 2015.

[8] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[9] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. *CoRR*, abs/1406.2199, 2014.

[10] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[11] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.