Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS 2020)
IEEE Xplore Part Number:CFP20K74-ART; ISBN: 978-1-7281-4876-2

Genre Classification of Movie Trailers using 3D Convolutional Neural Networks

[1]Prashant Giridhar Shambharkar,
Department of Computer Science & Eningineering
[1]Delhi Technological University, Delhi
prashant.shambharkar@dtu.ac.in

[1]Pratyush Thakur,
Department of Computer Science & Eningineering
[1]Delhi Technological University, Delhi
pt05@live.com

[1]Shaikh Imadoddin
Department of Computer Science & Eningineering
[1]Delhi Technological University, Delhi
furqan9196@gmail.com

[1]Shantanu Chauhan
Department of Computer Science & Eningineering
[1]Delhi Technological University, Delhi
schauhan.cshantanu@gmail.com

[2]Dr. M N Doja
Department of Computer Engineering
[2]Jamia Millia Islamia, New Delhi
mdoja@jmi.ac.in

***Abstract*— There is a constant influx of movies every year in the entertainment industry. A crucial ingredient that helps movies succeed at the box office is its trailer. A trailer helps a viewer form an opinion about the movie and is often an accurate representation of the movie. Thus, genre identification of movie trailers is a significant problem that can be solved for movie categorization and censorship. In this paper, we propose using a 3D CNN for solving this problem. The advantage of 3D CNN is its ability to capture the spatial as well as the temporal information present in a movie trailer. We achieve an accuracy of 82.14% which is greater than other models that utilize just 2D CNNs.**

***Keywords*— 3D CNN, movie, trailer, genre, identification, spatial, temporal.**

I. INTRODUCTION

Movies are an expression of art that caters to the entertainment of the human race. A movie is identified by its trailer, and therefore trailers can be compared to the blueprints of the movie. An effective model for the genre-based classification of movie trailers based on three-dimensional convolutional neural networks is proposed in this paper.

The techniques to classify movie trailers may be classified into the following three primary methodologies- based on text, based on audio and an approach based on the video. The importance of a trailer lies in the ability of a trailer to garner the attention of the audience before a movie is even released. A trailer showcases the best parts of a film, to convey to the audience what they can expect in theatres.

An important aspect of trailers or even a movie is its genre. A very common problem that has been discussed widely in scientific literature is the classification of movie trailers based on its genre. To tackle this issue, the use of a 3D CNN is proposed that takes into account the spatial features as well as the temporal features present in a trailer. We download the trailers from YouTube, the details of which are discussed further in the paper and use tools like FFmpeg to downsample the frames present in the trailer. The input stacks 64 frames at a time, across 3 colour channels for feature extraction.

The use of 3D CNN is lucrative because of one simple reason, i.e. its ability to capture information in the temporal field. It can extract information that a 2D CNN couldn't, which leads to an increase in the performance of the model. Experimental results show that 3D CNNs can outperform normal 2D CNNs. This classification of movie trailers into genres has multiple use cases and advantages, such as a viewer knows what to expect when he will go to see a movie and also it can help in the efficient screening of movies based on the content of the movie. The classification may also help in attributing genre tags on websites like IMDb or Netflix.

The rest of the paper is organized in the following fashion: firstly the previous work done in this field is discussed in Section 2. Then, the methodology that we

propose is discussed in Section 3. The results obtained and comparison with 2D CNNs is discussed in Section 4. Lastly, the conclusion of the paper is presented in Section 5.

II. LITERATURE SURVEY

A. Classification based on audio-visual features

Earlier techniques in this field pertain to extracting important low-level features and then making sense of these features to help classify the genre. There are a variety of features that are extracted in different research papers and different techniques used for classification. All of these are discussed below-

In papers [4], [5], [6] and [8], the emphasis is given on video-based genre classification. [4] presents us with the way of automatically classifying the video into different genres based on its content. It uses many audio-video cues as well as structural knowledge so that the video is differentiated into various genres. This system has been incorporated into content-based video processing systems and has also been useful for the detection of high-level features in TRECVID evaluations. This system has also been evaluated based on YouTube videos and French channels having an accuracy of 92.4% and 99% respectively. This provides a good way of classifying the videos paving the way for classifying the trailers on YouTube.

[5] tells us about the computational characteristics studied from effects through editing, motion, and different color used in videos to classify videos into different categories. These characteristics apart from taking human understanding to classify the videos also use the rules set by some directors to back the specific features to different genres to create an impact on viewers. [5] also deals with the issue of the specific duration required to get a desirable genre classification. These sets of characteristics have the visual features of a video sequence to classify it into different genres like sports, news, commercials, and cartoons. Temporal sequencing of shots and their semantics is missing in this proposed system that can improve its efficiency.

[8] tells us about the technique to automatically segregate the video shots into different categories. This method uses the incoming video sequences to classify the video sequences into different categories. [8] discusses two methods for automatic classification. The first method analyzes the video sequence to extract motion information and thus uses this information to train Hidden Markov Models (HMMs) that classify the videos with greater precision. The results showed the classification accuracy of around 90% by the HMMs. The second method classifies 24 full-length motion pictures using HMMs. The classification had an accuracy of 91.67%. This system lacks to take into account the viewer's perspectives to see what a viewer likes or dislikes. Classifying video sequences by genre helps to narrow down the search but it is not necessary that the viewer likes all the movies in the particular genre.

[6] also discusses the automatic video genre classification. [6] additionally proposes three different content descriptor classes. The information contained in a video at the temporal level is described by the optic flow, the action content, and the number of continuous changes. One level further, color distribution statistics, elementary complexion, different properties, and relationships are used to describe colors. At the final level, we extract the structural information at the level of the image as well as the contours, and information is explained by building histograms. The average correction ratio for detection is up to 95% whereas the recall and precision ratios are up to 100% and 98% respectively. Some drawbacks such as audio information have not been incorporated into this system. More video genres could have been tested.

[1] proposed a system to classify movies by manipulating the audio-visual features of previews. A preview can be simplified as a summary version of a movie. In this approach, a movie is divided into two categories specifically into action or non-action by the calculation of average shot length as well as the visual disturbance of each video clip. Then cinematic principles in conjunction with audio and color principles are used for classification of movies into horror, comedy, or movies that have violence. This system is also used to browse and retrieve videos on the internet, maintain video libraries, and to rate movies. The semantics could have been explored more from shot level to scene level. Less noisy frames should also be used.

[2] presents a technique to classify the movies based on scene categorization. [2] uses a high-level semantic comprehension of movie clips rather than just utilizing low-level features. [2] proposes that the keyframes be grouped into a collection by using an algorithm that analyses the shot boundaries. After this, it uses modern detectors and descriptors namely GIST, CENTRIST, and W-CENTRIST that obtain features, which are then utilized to categorize scenes.

Then the trailers are represented by the bag-of-visual-words model (bovw) which contains shot classes as its vocabulary. This bovw model is mapped temporally to four genres of movies namely comedy, action, horror, or drama movies. This improves accuracy in classifying movies as compared to just extracting low-level visual features. But this system has certain drawbacks. Dynamic components are not taken into consideration. Movies that are black and white have quite different scene features from those which are colored. Hence, making them difficult to categorize.

[3] and [7] are used for film classification using trailer features and computable features respectively. [3] discusses the features that can be extracted from the trailer so that it helps in the film classification. By using the trailer dataset. [3] aims to compute the category of genre and also the MPAA rating of a film by the use of video features and subtitle texts. Three main types of video features are used namely mean frames, scene variation, and scene categorization to cut back the data into an easy to handle set. In text-based classification, the subtitle file was fed as input to a matrix in which each row consists of one sample and every column is a word. The classification is based on 3 main algorithms namely the random forest algorithm, multi-class support vector machine, and naive bayes algorithm. This method should incorporate the testing of a larger database instead of smaller ones. Along with video features and subtitles text, audio features could also have been added.

[7] brought to the fore a framework which classified the genre of movies based on estimating four low-level features from the previews of the movie. They hypothesized that all filmmakers follow some rules and based on 'film grammar rules', one could establish a relation between the symbols of grammar and the computable characteristics. These characteristics include the motion content in a trailer, the average shot length, the color variance in the clip, and the lighting key. Based on these, further classification of the previews into genres was done. A mean shift algorithm is put to use in this method to categorize movies. This approach can also be applied to update databases, scene understanding, recovery, and browsing of videos on the internet. One issue with this method is to automatically classify videos and the relevant field indexing and annotation. Also, their work doesn't take care of the fact that not all features of a picture can be converted to a statistic for example emotions, surprise, and shock.

[9] introduced a decision-based classifier based on low-level features. It incorporated YUV color space which is used for television transmission and HSV color space which is used in computer graphics. It makes its decisions based on two features- decision using visual effects and decision using light. The visual effect features include two types namely slow-moving effect and fast-moving effect. However, the dataset consisted of only 44 films and the classification resulted only in three broad categories.

[10] designed a neural network that classifies movies based on different genres. Along with low-level video features, it also used audio features. Both of these features, which are 21 in nature are fed into the neural network which contains neurons arranged in a single layer. Based on the input, the neural network outputs one of the five genres which are comedy, horror, music, action, drama. However, the absence of any hidden layers gives rise to the conclusion that this is another type of logistic regression.

[11] came up with an approach that uses the SAHS(Self Adaptive Harmony Search) algorithm in selecting features for different movie genres. After this, a Support Vector Machine is loaded with a set of features which consists of 277 features extracted from every movie trailer. After this, they used a majority voting technique to estimate the genre of the movie. They make use of both visual as well as audio characteristics in this technique and use a maximum of 25 features to distinguish between two different genres of movies.

B. Deep Learning Techniques

The rise of machine learning has given new methods of solving conventional problems. Deep Learning is a specialized field in the realm of Machine Learning that is increasingly becoming a preferred approach in solving classification problems. Convolutional Neural Networks belong to a class of networks called Artificial Neural Networks that can solve classification problems related to images very efficiently. Also, since a movie is nothing but some frames placed one after the other, there has been quite some success in using convolutional techniques to classify genres.

[12] introduced the method of using Convolutional Neural Networks for figuring out high-level visual features. Since a movie trailer is nothing but a sequence of frames, they proposed the use of CNNs for identifying the genre of a movie by its trailer. The network was trained frame by frame

using the LMTD-4 dataset. The features so extracted were then put together to help create semantic histograms of various scenes. Alongside these, audio features were combined, and then the whole representation of each move trailer was fed into SVMs to classify it based on its genre. However, sometimes using the multi-label loss function used in the approach can be instrumental in preventing the Convolutional Neural Network from converging. This may happen to owe to a phenomenon called the vanishing gradient problem and might significantly impact the accuracy of this method.

[13] proposed a relatively new architecture called Convolution-Through-Time for Multi-label Movie genre Classification (CTT-MMC). The architecture consists of a deep neural network that makes use of the traits from the trailer of different movie frames across time. His module uses convolutions to learn the spatial as well as temporal characteristic based relationships of the entire movie trailer. Using a two streamed based strategy he studied both audio and video features. A 152-layer CNN which is residual is used to extract high-level frame-based information. Though the idea of using temporal relations makes this approach faster and less error-prone than LSTMs, the temporal analysis is performed only after the spatial features have been extracted, but in the process of spatial features extraction, a lot of temporal information is lost resulting in a less efficient network.

[14] worked on a neural network based on the VGG16 framework. A stacked LSTM model was implemented; considering a time step of 9, Like the spatial approach, the input to the LSTM was VGG features. An accuracy of 80.1% with the spatial features and 85% with using LSTM was achieved.

[15] used a deep neural network for achieving the classification of movie genres using the poster of a movie only. [16] proposed that ResNet34 and a custom architecture are reasonably successful at predicting a poster's genres. Though these papers use a resnet, where it is used only as a 2d-CNN. The use of pseudo-3D CNN and 2+1D resnet isn't done on movie trailers which can be much better.

[17] proposed two methods capable of handling full-length video. While adopting CNN for this task to examine various design options that examine different design options, the first approach explores different homogeneous temporal feature pooling architectures. The second proposed technique expressly depicts the video as an arrangement of frames in an ordered fashion. For this reason, a recursive neural network using LSTM cells that are connected to CNN's output is proposed.

In another approach suggested by [18], movies are classified into different genres like horror, comedy, drama, etc using CoNNeCT (Convolutional Neural Networks for Classifying Trailers). [19] considers that the different genres of movies are intangible. They present a hypothesis that multiple convolutional neural networks that are trained to learn different features of the movie scenes such as object detection, motion content also perform the classification into different intangible features. 5 different CoNNeCT models were proposed and the predictions from each of the five models were concatenated.

[19] proposed a multimodal KDK classifier for classifying movie trailers. They took into consideration both the audio as well as video features. They introduced a new video feature as well as three new audio features which proved useful in classifying the genre. The combination of CNN with audio features gives promising results. They achieved an accuracy of 81% in predicting the most important genre and an accuracy of 91% while predicting the two most important genres of the trailers present in their dataset.

Despite 3D CNNs being computationally expensive, many approaches have explored them due to the added benefits of exploring the temporal dimension along with the spatial dimension.

[20] proposed a generic Convolution 3D feature which they refer to as C3D. This feature was obtained by training a deep three dimensional CNN on a massive data set that consisted of scenes, actions, objects, and other features. They demonstrated that this feature is compact, efficient to compute, and achieved results that were state of the art.

[21] proposed a 3D convolutional neural network called C3D, training it on a dataset that was supervised, leading to the inference that 3D CNNs are much better for extracting and learning spatiotemporal features rather than using conventional 2D CNNs.

[22] introduced a novel method for human action recognition using 3D CNN. They constructed a 3D architecture that would generate information channels from the subsequent frames of the video being analyzed. The information so discovered would be then put together to create a final feature representation of the input data. They also regularised their models for boosting the performance.

[23] showcased the use of three-dimensional convolutional neural networks for sign language recognition. They used Microsoft Kinect for providing input to the model. The input had 5 channels which included depth as well as body skeleton in addition to the standard color information present in the RGB channels. Nine such frames were stacked together across these five channels to be fed at a time in this network. The model outperformed the baseline method for sign language recognition.

Thus, 3D CNN seems promising in its application for solving video classification problems despite the inherent problems of the computational expense involved. In our problem of genre classification, using a 3D CNN seems like a fruitful approach.

III. METHODOLOGY

Convolutional Neural Networks have a place with a class of neural systems that perform exceptionally well for studying and classifying images as shown in Figure 1. CNN takes images as inputs, and its structure is similar to a traditional neural network with neurons connected in layers, which subsequently have weights, biases, and activation functions associated with them. A neuron takes some input, performs some computation using the activation function, and passes on the data.

In essence, CNN uses its filters for finding out the features present in the image and then using the most important features for prediction.

The reason why CNNs perform better than simple feed-forward neural networks is the ability to better understand spatial as well as temporal relations due to the application of useful filters. The parameters used are reduced, and weights can also be reused to achieve better results.

In 2D ConvNets, we perform convolutions to calculate and compute characteristics only from the spatial dimensions. The kernel will move over the data it receives from the layer preceding it and outputs a reduced feature map. The parameters of the kernel remain the same as it moves over the entire data from the layer preceding it. This is known as weight sharing. The major benefit is that this sharing is instrumental in the reduction of the count of free variables. The network's generalization capacity is also increased.

In 3D ConvNets we apply different operations at each location of input so that we get different features which include both the dimensions- spatial as well as temporal dimensions. The kernel that we use is three dimensional. It convolves across the cube formed due to frames being stacked together.

A. CNN Architecture-

The CNN architecture consists of multiple layers, usually comprising of the following components to increase its efficiency- Convolution Layer, Pooling Layer and Fully Connected (FC) Layer

1) Convolution Layer: This is the first layer that is used to select and abstract information from the input image. It is also the major building block of the convolution neural network. It contains different sets of filters whose parameters are needed to be trained. Convolution is similar to a mathematical operation that takes two filters or kernels as inputs. The output volume is calculated by applying this mathematical operation between the kernels and the part of the picture. If we are using an image of width 32, height 32, and depth 3 making it a 32x32x3 dimension image and applying 12 filters for this layer, we will have an output of dimension 32x32x12. This output results in an activation map made of different neurons. In some of the cases, the different neurons share the weights in the activation map. This is called parameter sharing. Also in local connectivity, some of the neurons are associated with a small part of the picture that is input. Local connectivity helps in reducing the different sets of parameters used. ConvNets learn multiple features in parallel instead of learning only a single filter. This helps in the abstraction of features with a very high order as we go deep in the neural network.

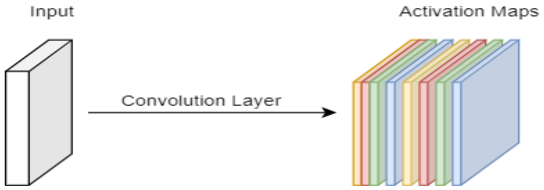

Fig. 1. A Convolution Layer

2) Pooling: Pooling can be called as a process that merges data. Its purpose is to reduce the data size. The significant role of this layer is to lessen the spatial size of the portrayal by decreasing the number of parameters, variance, and calculation in the network. Low-level highlights can be extracted along with a set of smooth features. Reducing the dimensions helps in fast computation and also helps in

preventing overfitting. Two kinds of pooling layers are max pooling and average pooling. A sample-based discretization process can be called max pooling. Max pooling is carried out by using a max filter or kernel to non-overlapping sub-regions of the internal representation. An average pooling layer performs down-sampling by separating the contribution to rectangular pooling locales and registering the average estimations of every region. It takes into account all the values for feature mapping and output creation.

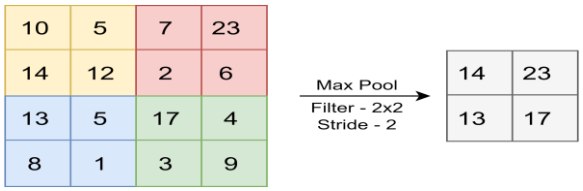

Fig. 2. Max Pooling Operation

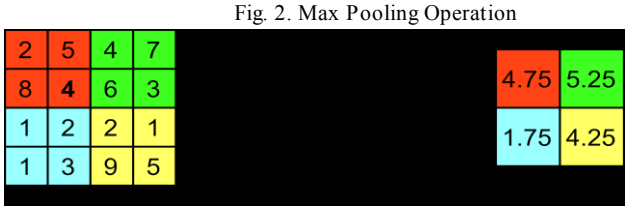

Fig. 3. Average Pooling Operation

3) Fully Connected (FC) Layer: It is a simple feed-forward network layer. In this layer, each cell or neuron is connected to each of the elements of the previous layer and receives the input. The important task of classifying the image into different labels by analyzing the results of the convolution and pooling lies with this layer. The output from the pooling process is flattened and is input to this layer. The convolutional layers give us a significant, low-dimensional, and a little invariant component space, and the fully-connected layer is learning capacity in that space. The last fully-connected layer is utilized to find the probabilities that the image has a place within a specific class (classification) through the help of the softmax activation function.

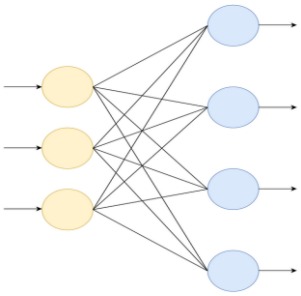

Fig. 4. Fully Connected Layer

B. Proposed Architecture

The use of a 3D-CNN gives us the advantage of learning temporal as well as spatial features from the video frames. Movie trailers are simply frames that are stacked together one after the other.

In our proposed architecture, convolutions take place over 64 frames that are stacked together. The filters will convolve over the volume and learn features that will be utilized for classification. It will start by learning rudimentary features like simple edges. As further convolutions occur, the model will learn high-level features which will help in the classification of movie trailers.

The benefit of using 3D convolutions in our architecture is the capturing of temporal information from the frames. For example, if a gun is fired in a trailer, many succeeding frames will have the trajectory of a bullet. A network can recognize such an occurrence only if sufficient frames are available together. In 2D convolutions, there's only one frame and hence, such extraction of information is not possible. Thus, scenes like these may further help the network to learn that this trailer may belong to the action genre since a gun is being fired.

For efficient implementation, we configure the data-loader so that the consecutive frames are fed to the underlying network properly for the network to analyze our data. In our approach we use a sliding window technique, taking the window size to be 64 frames. Therefore, the 64 consecutive frames of the movie trailer are being fed and subsequently examined by the 3D Convolutional Neural Network. A step size of 32 frames is chosen to help implement the sliding window.

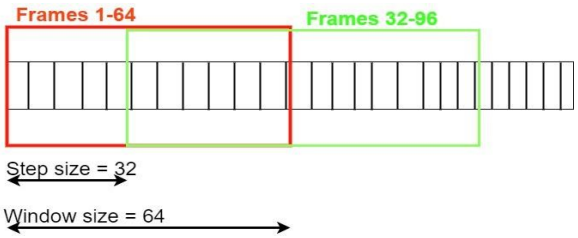

Fig. 5. Sliding window technique implemented in data loader

For example, a trailer having 128 frames, the first series of frames include frames numbered from 1 to 64. The second series will include frames numbered from 32 to 96. The third being 64 to 128. The whole trailer is processed in this fashion with an increment of 32 frames after each step. For example, a trailer having 128 frames, the first series of frames include

frames numbered from 1 to 64. The second series will include frames numbered from 32 to 96.

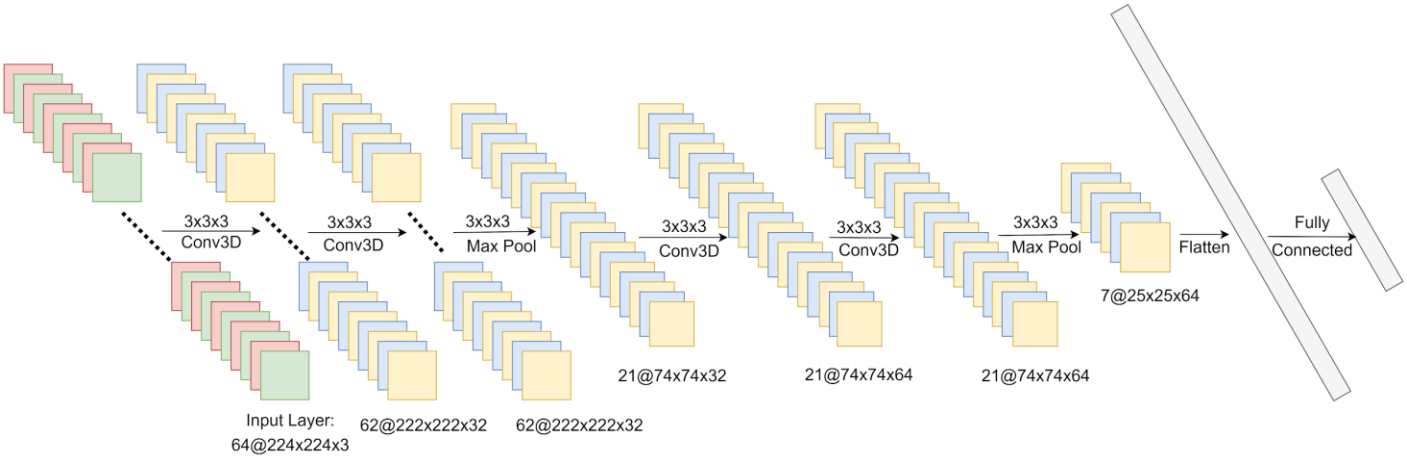

Fig. 6. Proposed architecture

The third series is 64 to 128. The whole trailer is processed in this fashion with an increment of 32 frames after each step.

The parameters window size and step size could be changed to achieve even better results, but to do so, there is a requirement of systems with powerful GPUs and RAMs preferably greater than 32 GB.

We propose an architecture that implements 3D convolutions as shown in Fig. 6. Our model consists of 8 layers. Our input consists of 64 frames of size 224x224 stacked together and then subsequently fed into the convolutional neural network. The beginning layer in our network is a convolution layer which has a ReLU activation, followed by another convolution layer with a softmax activation. It is then trailed by a maximum pooling layer. ReLu is used because of the computational efficiency it introduces coupled with nonlinearity. The softmax activation helps in a faster convergence during training. It constrains the 3D Network and forces it to be specific with its activations. It is again trailed by two subsequent convolution layers having ReLu and softmax activations respectively and a resulting max-pooling layer. This is then followed by two fully-connected layers that incorporate the output layer. A softmax activation function is used in the output layer.

TABLE 1. STRUCTURE OF OUR 3D CNN

Layer	Type	Filters	Kernel Size
1	Convolution	32	3x3x3
2	Convolution	32	3x3x3
3	Max Pooling	-	3x3x3
4	Convolution	64	3x3x3
5	Convolution	64	3x3x3
6	Max Pooling	-	3x3x3

C. Algorithm

1. In the data loader, the images are resized to 224 x 224.
2. 64 frames are appended together in a sliding window fashion, as shown in figure xx, giving an input shape of 64 x 224 x 224 x 3 where 3 is the RGB colour channel for the frames.
3. Finally, when the input data is ready, it is sent through the 3D CNN model to learn and extract the spatiotemporal features.
4. Afterward, the features extracted are flattened and input to a fully connected layer that follows after it with activation function as sigmoid.
5. After that point, the fully connected layer is further connected with the output layer having the softmax activation function. This output layer represents the genres our trailer is classified into.
6. The steps 3 to 5 are repeated until convergence.

IV. RESULTS AND COMPARISON

A. *Dataset*

The trailers have been downloaded from different playlists available on YouTube. Each curated playlist has movies of a particular genre. These playlists were taken from YouTube channels like Movieclips Trailers, JoBlo Movie trailers, KinoCheck International, and segregated videos based on different genres. The different categories involved in this work are action, comedy, horror, and romance.

The total number of movie trailers in our dataset was 390. We have seen that the same movie trailer is also found in multiple genres but for this work, we have ensured that each trailer belongs to a single genre so that the movies belong to the particular genre they find a resemblance to. We used FFmpeg to downsample these trailers as a data preprocessing step. We extracted 1 frame for every second, excluding the first and last five seconds.

B. *Training and Testing*

We developed our Convolutional Neural Network model using Keras library in Python. Our training set is composed of 37460 sample frames of 4 classes i.e our genres-action, comedy, horror, and romance. Our test data is composed of 12846 frames. The images were resized to 224 x 224 using OpenCV. This training was done on Google Colaboratory with each epoch taking about 5 minutes.

We evaluated our model on our training dataset and the model was trained for 60 epochs. The optimizer used was SGD using a learning rate of 0.0001 and momentum 0.9, the loss function used was Sparse Categorical Cross Entropy function. The batch size was set to 2 because each of our samples was processing 64 frames simultaneously. Through this setup, we got a training accuracy of 99.47% and a testing accuracy of 82.14%.

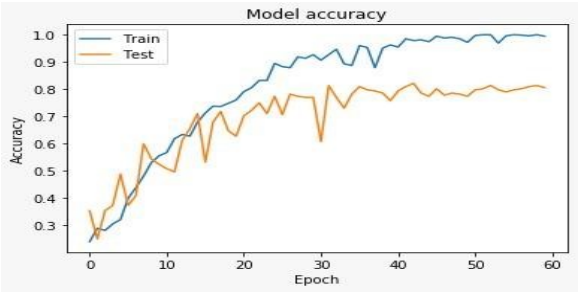

Fig. 7. Our Model Accuracy Graph

C. *Comparison with VGG16 and Inception V3*

We trained the VGG16[24] CNN architecture on the same dataset, this gave us an accuracy of 67.336. Evidently. the use of a 3D CNN outperforms it.

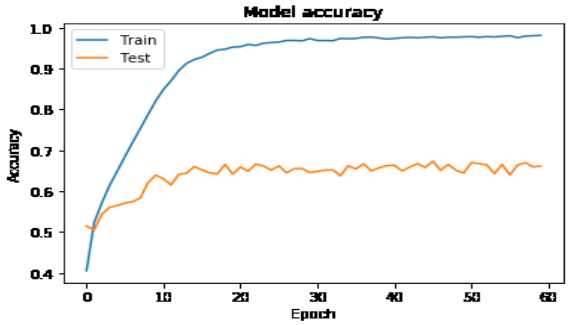

Fig. 8. Model Accuracy Graph for VGG16

We also trained the Inception V3[25] CNN architecture on our dataset, this gave us an accuracy of 67.648%.

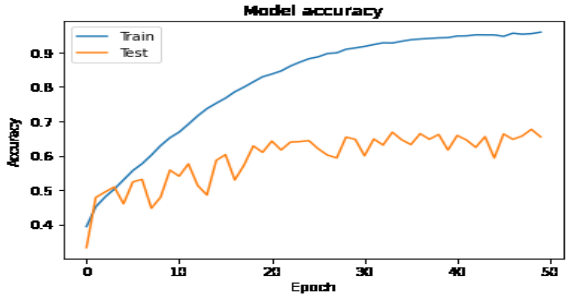

Fig. 9. Model Accuracy Graph for Inception V3

Comparison with these 2 models which are one of the best 2D CNN architectures for image classification to date shows us the importance of using 3D CNNs, as they take into account the spatiotemporal features, thereby achieving a much better accuracy of 82.14%.

TABLE II. COMPARISON OF ACCURACY WITH 2D CNNS

Architecture	Type of Convolutions	Accuracy(%)
VGG16	2D	67.336
Inception V3	2D	67.648
Our Approach	3D	82.143

V. CONCLUSION AND FUTURE WORK

We developed a 3D CNN model for movie trailer classification into namely four genres-action, comedy, horror, and romance. This model is capable of learning and extracting spatiotemporal features by the use of 3D convolutions. This architecture performs convolutions on 64 consecutive input frames to learn and extract the spatiotemporal features It achieves an accuracy of 82.14%. For comparison, we trained both the VGG16 and Inception V3 2D CNN architectures on the same dataset, where our method significantly outperformed them, clearly showing the advantage of using a 3D CNN for our problem of Classification of Movie Trailers into respective Genres.

In the future, we could try to approximate our 3D convolutions by 2D convolutions preceding 1D convolutions, thereby separating the temporal and spatial ordering into 2 different steps. This will have the added benefit of rendering the optimization more easily. With a decreased computational complexity, this could get better results and better classify movies.

REFERENCES

[1] Z. Rasheed and M. Shah, "Movie Genre Classification By Exploiting Audio-Visual Features Of Previews," *IEEE Transactions on Multimedia*, vol. 7, no. 6, December 2005.

[2] Z. Howard, H. Tucker, A.V. Karandikar, and J.M. Rehg, "Movie genre classification via scene categorization," ACM Multimedia, pp. 747-750, 2010.

[3] E. Helmer and J. Qinghui, "Film Classification by Trailer Features," 2012.

[4] H.K. Ekenel, T. Semela, and R. Stiefelhagen, "Content-based Video Genre Classification Using Multiple Cues," in AIEMPro'10, Florence, Italy, 2010.

[5] B.T. Truong, C. Dorai, and S. Venkatesh, "Automatic Genre Identification for Content-Based Video Categorization," *ICPR 2000*, pp. 230-233.

[6] B. Ionescu, C. Rasche, L. Florea, C. Vertan, and P. Lambert, "Classifying documentary music news and animated genres with temporal color and contour information," *International Symposium on Signals Circuits and Systems - ISSCS*, 2011.

[7] Z. Rasheed, Y. Sheikh, and M. Shah, "On the Use of Computable Features for Film Classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 1, January 2005.

[8] G. Iyengar and A.B. Lippman, "Models for automatic classification of video sequences," SPIE vol. 3312, 1997, pp. 216-227.

[9] H. Huang, W. Shih, and W. Hsu, "A film classifier based on low-level visual features," Journal of Multimedia 3 (2008) 465–468.

[10] S. K. Jain, "Movies Genres Classifier using Neural Network," Proceedings of the International Symposium on Computer and Information Sciences, 2009, pp. 610–615.

[11] Y.F. Huang and S.H. Wang, "Movie genre classification using SVM with audio and video features," R. Huang, A. A. Ghorbani, G. Pasi, T. Yamaguchi, N. Y. Yen, and B. Jin (Eds.), AMT, Vol. 7669 of Lecture Notes in Computer Science, Springer, 2012, pp. 1–10.

[12] G. Simoes, J. Wehrmann, R.C. Barros and D.D. Ruiz, "Movie genre classification with convolutional neural networks," International Joint Conference on Neural Networks, IEEE, 2016, p. 8.

[13] J. Wehrmann and R.C. Barros, "Movie genre classification: A multi-label approach based on convolutions through time," Applied Soft Computing, 2017 Dec 1; 61:973–82.

[14] K. Sivaraman and G. Somappa, "Moviescope: Movie trailer classification using deep neural networks," University of Virginia, 2016.

[15] W. Chu, "Movie Genre Classification based on Poster Images with Deep Neural Networks," MUSA 2017 Proceedings of the Workshop on Multimodal Understanding of Social, Affective and Subjective Attributes, Pages 39-45, Mountain View, California, USA — October 27 - 27, 2017.

[16] G. Barney and K. Kaya, "Predicting Genre from Movie Posters".

[17] J. Y. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond Short Snippets: Deep Networks for Video Classification," IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015.

[18] J. Wehrmann, R.C. Barros, G. Simoes, T.S. Paula, and D.D. Ruiz, "Deep Learning from Frames," Brazilian Conference on Intelligent Systems, 2016.

[19] P.G. Shambharkar, M.N. Doja, D. Chandel, K. Bansal, and K. Taneja, "Multimodal KDK Classifier For Automatic Classification of Movie Trailers," IJRTE, Volume-8 Issue-3, September 2019.

[20] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "C3D: generic features for video analysis," CoRR, vol. abs/1412.0767, 2014.

[21] D. Tran, L.D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks," IEEE International Conference on Computer Vision (ICCV), pp. 4489-4497, 2015.

[22] S. Ji, W. Xu, M. Yang, and K. Yu, "3D Convolutional Neural Networks for Human Action Recognition," IEEE TPAMI, vol. 35, no. 1, pp. 221-231, 2013.

[23] J. Huang, W. Zhou, H. Li, and W. Li, "Sign language recognition using 3d convolutional neural networks," ICME, pp. 1-6, 2015.

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

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. "Rethinking the Inception Architecture for Computer Vision," arXiv:1512.00567, 2015.