Movie Tags Prediction and Segmentation Using Deep Learning

UMAIR ALI KHAN [1], MIGUEL Á. MARTÍNEZ-DEL-AMOR [2], SALEH M. ALTOWAIJRI [3], ADNAN AHMED [4], ATIQ UR RAHMAN [3,7], NAJM US SAMA [5], KHALID HASEEB [6], AND NAVEED ISLAM [6]

[1]Department of Computer Systems Engineering, Quaid-E-Awam University of Engineering, Science and Technology, Nawabshah 67450, Pakistan
[2]Department of Computer Science and Artificial Intelligence, Universidad de Sevilla, 41004 Seville, Spain
[3]Faculty of Computing and Information Technology, Northern Border University, Rafha 76413, Saudi Arabia
[4]Department of Telecommunication Engineering, Quaid-E-Awam University of Engineering, Science and Technology, Nawabshah 67450, Pakistan
[5]Faculty of Computer Science and Information Technology (FCSIT), University Malaysia Sarawak, Kota Samarahan 94300, Malaysia
[6]Department of Computer Science, Islamia College, Peshawar 25120, Pakistan
[7]Faculty of Computer Information Science, Higher Colleges of Technology, Ras Al Khaimah Campus, Ras Al Khaimah 25026, UAE

Corresponding author: Umair Ali Khan (umair.khan@quest.edu.pk)

This work was supported by ERCIM Alain Bensoussan Fellowship Programme.

ABSTRACT The sheer volume of movies generated these days requires an automated analytics for efficient classification, query-based search, and extraction of desired information. These tasks can only be efficiently performed by a machine learning based algorithm. We address the same issue in this paper by proposing a deep learning based technique for predicting the relevant tags for a movie and segmenting the movie with respect to the predicted tags. We construct a tag vocabulary and create the corresponding dataset in order to train a deep learning model. Subsequently, we propose an efficient shot detection algorithm to find the key frames in the movie. The extracted key frames are analyzed by the deep learning model to predict the top three tags for each frame. The tags are then assigned weighted scores and are filtered to generate a compact set of most relevant tags. This process also generates a corpus which is further used to segment a movie based on a selected tag. We present a rigorous analysis of the segmentation quality with respect to the number of tags selected for the segmentation. Our detailed experiments demonstrate that the proposed technique is not only efficacious in predicting the most relevant tags for a movie, but also in segmenting the movie with respect to the selected tags with a high accuracy.

INDEX TERMS Tags prediction, movie segmentation, deep learning, transfer learning.

I. INTRODUCTION

The huge amount of multimedia data generated these days makes it an ordeal to envisage techniques which can automatically check the contents of multimedia data to ascertain their authenticity and classify them accordingly. Especially, retrieval of required information from multimedia data and assignment of appropriate tags largely depends on manual processing. Hence, the quality of the assigned tags follows a subjective criterion and varies from person to person. Our preliminary experiments [1] in this regard demonstrate that human-generated meta data can not suffice to give full insight into the main contents of a movie and/or shows inconsistency due to the lack of precision in human's ability of information recall. In addition, manually-generated semantic tags are less accurate and present irregularities. Our preliminary experiments on this topic further reveal that this ostensibly trivial task entails an intelligent analysis of a video to predict its representative tags without human intervention. This automatically extracted information has immense applications in optimizing video search, automatically retrieving scenes from videos based on user's query, object detection and localization, automatic text/subtitles generation for videos, detecting specific events in videos, action recognition, behavior recognition, recommendation systems, etc. Among these applications, scene-driven retrieval is particularly important in the sense that it not only helps in content-censorship (e.g., automatically censoring the scenes containing nudity, sex, violence, smoking, etc), but also in on-demand retrieval of desired scenes from a given movie (e.g., making highlights of a soccer match which contain all the goal events). At the same time, scene-driven retrieval is equally important for video summarization, e.g., removing all boring or unwanted scenes.

Segmenting a movie in the major constituent topics does require a precise identification of these topics in the first place. This information can then be used for on-demand scene selection, content-censorship and other tasks. In this paper, we address the problem of predicting key information from a movie in the form of small number of tags which describe the overall contents of the movie. For this purpose, we aim not only to understand the semantic meaning of individual movie frames, but also to predict a compact set of the movie's representative topics. The predicted information can be utilized in a number of ways such as movies categorization, context-based search, content-censorship (e.g., nudity, violence and sex in kid movies). Apart from this, the predicted tags for a movie can be further utilized to segment the movie according to the user's choice.

Applying the traditional approaches of object detection on the individual movie frames will be inefficient as we will end up with low-level information (e.g., the localization of objects in the individual frames instead of the contextual relationship among them). In addition, processing each movie frame will also introduce computational inefficiency and result in redundant information. This problem can only be addressed by a machine learning algorithm which can be trained to predict the high-level, representative features of movie scenes. The tremendous advancements in the field of machine learning have paved the way for finding patterns in complex data with an accuracy which, in some cases, even surpasses human's pattern matching performance. The cheap and scalable parallel processing technique utilizing Graphical Processing Units (GPUs) have made possible to efficiently apply machine learning techniques for image/video analytics [2]. That said, applying machine learning to learn the traditional image features for our problem is not efficient due to the well-known issues related to these features such as requirement of mathematical modeling, limited generalization, scale- and rotation variance, inability to maintain performance under different conditions, etc.

Instead of learning the hand-crafted image features, it is more efficient to discover the underlying features in the individual movie frames. Deep learning [3] can serve this purpose, as it does not require a priori information of image features. Instead, it learns the underlying patterns in complex data during training. Apart from this, a deep learning model trained on a large dataset can be retrained using transfer learning [4] for a different classification task with a much smaller dataset and training time. Considering the promising features of deep learning, we formulate the problem of movie tags prediction as a deep learning based classification. For this purpose, we first develop a tag vocabulary in which each tag represents a class. We further develop a dataset corresponding to each tag in the vocabulary. Subsequently, we transfer the features of a pre-trained Convolution Neural Network (CNN), Inception-V3 [5], for our training task by modifying and re-training its final layer using transfer learning.We further propose an efficient shot detection technique for determining the key frames in a movie which are later used for analytics by the deep learning model. The proposed shot detection technique is able to detect the hard-cut, fade-in and fade-out shot boundaries.

Once all the key frames in a movie are found, their CNN features are computed and fed to the newly added final layer of the trained model to get the corresponding predictions for each frame. The predictions corresponding to each frame are assigned certain scores and relative weights based on the values of their prediction probabilities and predominance in the movie. The tags having smaller relative weights are dropped out to obtain a compact set of few prime tags which best represents the overall contents of the movie. Discarding the motion information by processing only the key frames of a movie does not have a considerable impact on the prediction accuracy, as the recent related work in this domain reveals that motion features do not drastically impinge on this task [6]–[8].

The key frame analytics by the deep learning model further generates a corpus which is used to segment a movie with respect to a selected tag. The corpus contains the details of each shot's boundary, its key frame, the predicted tags for the key frame, and the spatio-temporal details of each shot. We further analyze the segmentation performance with respect to the number of predictions per key frame selected for the segmentation. The results presented in the paper show the tradeoff between the precision and recall of the segmentation.

The conspicuous features of the work presented in this paper can be summarized as follows.

- The proposed technique of movie tags prediction operates at a higher semantic level by seizing the overall context in the extracted key frames of a movie. The context denotes the semantic meaning of the inter-objects relationship in a scene, for instance, violence, action, romance, fight, etc.
- This work stands apart from the traditional event/scene recognition approaches where each item is adherent to a single event or scene.
- Our proposed technique is also in contrast with the traditional object recognition techniques which targets to localize and label every individual object in an image. This will result in a highly redundant and inconsequential information for our task.
- This work does hold resemblance with genre classification of movies. However, a movie typically has 2-3 genres which do not encompass the entire range of a movie's contents (e.g., nudity, sex, smoking, violence, etc). In contrast, our carefully designed, flexible and scalable tag vocabulary sufficiently covers the main theme of a movie.
- Unlike the existing video segmentation techniques which perform little to no semantic analysis and mostly exploit the visual similarity of the shots to merge them into non-overlapping scenes, we do

consider a shot's semantics to label and categorize it accordingly.

The rest of the paper is organized as follows. Section II provides an overview in the domain of movies/videos tags prediction and segmentation. Section III provides a brief theoretical background of convolution neural networks and transfer learning. Section IV describes our movie tags prediction and segmentation algorithms in detail. In Section V, we discuss the experimental setup and evaluation results of movie tags prediction and segmentation. Section VI concludes the paper.

II. RELATED WORK

We believe that movie tags prediction and segmentation has not been well-studied in the literature. The related work in this domain is primarily targeted at video tagging on a limited scale. Qi *et al.* [9] annotated certain concepts in a video using multi-label classification and the inter-class correlation. Another video labeling approach proposed by Siersdorfer *et al.* [10] put to use the redundancy among YouTube videos for finding associations among videos and assigning tags to similar videos. The techniques proposed by Shen *et al.* [11] and Liu *et al.* [12] made use of the data captured by the smartphone sensors to generate video tags. In Miranda-Steiner [13], the proposed technique identified basic objects in the images and videos of a digital camera and further exploited the geographical and date/time information to predict the relevant tags.

Ulges *et al.* [14] first found the key frames from a video and then predicted several visual features for each key frame. The visual features were assigned scores which were later fused to generate a final probability for a certain tag in the video. The tagging performance in this approach largely depended on the feature modalities and thus had limited accuracy. Chen *et al.* [15] proposed a video tagging technique which first found all the textual descriptions of a video from Internet sources and a graph model was applied on the descriptions to discover and score the key words serving as tags. This technique was dependent on the human-generated textual description. In a similar technique, Zhao *et al.* [16] first found similar videos by local features. The tags from the similar videos were analyzed to pick the most relevant tags for the given video. It is palpable that this technique shared the same limitations as in [15].

Some techniques proposed by Aradhye *et al.* [17], Toderici *et al.* [18] and Yang and Toderici, [19] did not solely rely on the user-supplied tags, but also took into account the audiovisual features to train classifiers based on the correspondence between the contents and the user-annotated tags. Nevertheless, the incorporation of inconsistent user-supplied meta data introduced the aforementioned issues.

A large part of the relevant literature in this context relied on the user-annotated meta data. A similar technique proposed by Chu *et al.* [20] first searched for the images on Flickr that has similar tags as those of the given video. A bipartite graph was used to describe the relationship between the key frames of the video and the tags associated with the images. The technique proposed by Acharya [21] selected one or more user-generated tags for a video which described its category. A transcript of plurality of words was generated along with their respective ranks. Based on the ranking of the plurality of words, one or more tags were generated. Chen *et al.* [22] proposed a web video topic detection technique. This technique utilized the video related tag information to determine bursty tag groups based on their co-occurrence and temporal trajectories. The near-duplicate key frames predicted from the web videos were fused with these tag groups. Subsequently, the fused groups were further matched with the keywords obtained from the search engine to find the topics.

Some techniques [23], [24] used the plot synopses and summaries of movies to predict a set of tags or movie genres. These techniques required plot synopses of movies which are not always readily available with a movie. In addition, the dataset was comprised of manually curated tags which share the same aforementioned limitations.

Ullah et al. worked on video semantic segmentation for pedestrian flow and crowd behavior. In [25], they identified crowd behaviors from a video sequence using a method based on thermal diffusion process and social force model. In [26], they employed a recurrent conditional random field using Gaussian kernel features to segment anomalous entities in pedestrian flows by detection and localization. In a recent work [27], they proposed the hybrid social influence model for pedestrian motion segmentation by using a particle representation and modelling the influence of particles on each other. However, these methods are domain-specific and focused on segmentation inside the images of the frames.

Due to the recent breakthroughs and advancements in deep learning, the research on semantic analysis of images and videos has been diverted to use complex neural architectures to learn hierarchical feature representations. The hot research areas in this domain include converting visual data to textual representation [28]–[31], answering questions from videos [32], [33], and video classification [7], [34]–[36]. The first two areas are different from tags prediction, as they entail more sophisticated architectures such as recurrent neural network [28] in combination with Convolutional Neural Network (CNN) to discover the spatio-temporal connection between consecutive video frames. On the other hand, video classification does hold resemblance with video tagging, but it is mainly focused on predicting the major category a video falls in, rather than predicting an extended set of classes pertaining to a given video.

A number of video segmentation techniques have been studied in the relevant literature. Majority of these techniques use a common approach: finding the shot boundaries and merging the shots into uncategorized segments (scenes) based on their visual similarity. A shot is an elementary structural segment that is defined as a sequence of images taken without interruption by a single camera [37]. Rasheed and Shah [38] clustered the shots based on their color similarity and found the segment boundaries based on the shot lengths and the motion contents. Some techniques [39]–[41] addressed the video segmentation by constructing a shot similarity graph using the color and motion information and subsequently segmenting the video by graph partitioning. Some shot clustering techniques [42], [43] also used Markov chain Monte Carlo technique for detecting segment boundaries, albeit the segments were uncategorized. In another shot clustering approach, Chasanis *et al.* [44] applied a sequence alignment algorithm to detect the change in pattern of shot labels to determine segment boundaries. In another technique, Chasanis *et al.* [45] first found the local invariant descriptors of the key frames of all the shots and grouped them into clusters. Each cluster was treated as a visual word. The histograms of visual words were smoothed using Gaussian kernels whose local maxima represent the segment boundaries. In a different approach, Hoai *et al.* [46] augmented video segmentation with action recognition. They first trained a recognition model using multi-class SVM on a labeled dataset. The segmentation and action recognition was done simultaneously using dynamic programming. This was the first approach of video segmentation with segment categorization, though in the form of limited number of action recognition. However, it did require the engineered image features for training the action recognition model.

Some approaches combined audiovisual features for scene segmentation. Sidiropoulos *et al.* [47] addressed this issue with a semantic criterion by exploiting the audiovisual features of the key frames to construct multiple Scene Transition Graphs (STGs) [48]. A probabilistic merging process combined the results of the STGs to detect segment boundaries. In a similar approach, Bredin [49] extended this idea by combining speaker diarization and speech recognition with visual information. A drawback of these techniques is that the STGs exploit low-level visual features and provide no margin for augmenting heterogeneous feature sets. In addition, the heuristic settings of certain STG parameters are also required.

In another technique, Baber *et al.* [50] used frame entropy to find shot boundaries and determine the key frames of the shots. Afterwards, the SURF features of the neighboring key frames within a window were matched to determine the scene boundary. In a later approach Baber *et al.* [51] the histogram of visual words for each shot were computed. The distance between the visual word histograms was calculated to merge the shots which are closer in space.

In a more recent approach, Yanai *et al.* [52] found the relevant shots from web videos based on the given keywords. This technique first searched for the relevant web videos by matching their human-generated tags with the given keywords. It then segmented the selected videos into shots and ranked them according to the similarity of visual features. The top-ranked shots represented the shots of interest.

A detailed literature review in this domain reveals that video tagging and segmentation has not been studied in combination. Whereas the existing techniques of video tagging either depend on hand-crafted image features and user-annotated meta data or do not provide an extended set of the thematic points of a movie, the semantic criteria in the video segmentation is largely ignored. The commonly used approach of matching the low-level visual and/or audio features of the successive shots (or their key frames) to determine the segment boundaries is too trivial to understand the semantic correlation among the shots. Additionally, segmenting and merging all the logical story units based on the semantic understanding of individual shots can not be efficiently done by low-level, engineered audiovisual features. Hence, segmenting a video into constituent topics, which can be later retrieved by a query, requires an intelligent semantic analysis of each shot. This is only efficiently possible by a deep learning based algorithm which does not require a priori knowledge of the low-level features.

We addressed this issue in a threefold approach: (i) we first proposed an efficient shot boundary detection algorithm which finds the representative key frames of all the shots in a movie, (ii) we trained a convolution neural network on a tag vocabulary to predict the context of each key frame and subsequently generating a compact set of the movie tags without requiring a priori information of image features or user-annotated meta data, and (iii) we offered an on-demand segmentation of the movie based on its predicted set of the tags. Using the semantic information provided by the movie tags, we eliminate the need of matching the low-level audio-video features of the successive shots for segmentation. Our segmentation approach classifies a shot into a particular category based on its contents. Hence all the relevant neighboring shots can be efficiently merged into a particular category. In this way, our movie segmentation approach is the first to use the semantic criterion for segmentation.

III. CONVOLUTIONAL NEURAL NETWORK (CNN) & TRANSFER LEARNING

Contrary to traditional neural networks, CNNs have much higher number of hidden layers which are well-suited to discover the intricate patterns in complex data without a given mathematical model. Due to this appealing feature, the last decade has witnessed a tremendous potential of CNN for semantic analysis of images and videos. A CNN has four types of layers: (i) convolution layer extracts features from a given image using multiple filters, (ii) activation layer restricts the output of the convolution layer in a specified range and introduces nonlinear mapping and generalization in the learning process, (iii) pooling layer reduces the spatial size of the data which results in less number of parameters and computations, and (iv) classification layer outputs a probability distribution which contains the final score of each class. A CNN architecture may range from simple (having relatively smaller number of convolution, activation and pooling layers) to complex (having hundreds of layers). The deep architectures help CNN to discover patterns/features in complex data without a given mathematical model.

The deep architecture also offers a dedicated challenge of training a CNN from scratch, as it requires enormous computing power, incredibly long training time, and a huge training data. However, analogous to human learning, the knowledge acquired by a CNN pertaining to a specific problem is transferable to another problem [53]. As we move from lower level layers of CNN to higher level layers (in the direction of classification layer), the specificity of features increases until the final classification layer becomes entirely task specific. The image features extracted by the lower level CNN layers can be utilized to re-train the model for an entirely different task, eliminating the need of training the model from scratch. In this connection, all the layers of a pre-trained CNN model, except the final classification layer, can be used as fixed feature extractor. The final layer can be modified and re-trained for a new task, utilizing the knowledge obtained from the previous training. This method is called transfer learning which we use to re-train a CNN model, Inception-V3, trained on a large dataset (ImageNet[1]). Although this CNN model has been trained for a completely different task, its features are effectively transferred to the task of movie tags prediction.

IV. MOVIE TAGS PREDICTION AND SEGMENTATION

To the best of our knowledge, there is no public dataset of movies containing labelled static scenes related to the training classes of our tag vocabulary. Hence, we first develop a tag vocabulary comprising of 50 movie tags. Subsequently, we construct a dataset for each tag by collecting the relevant features (movie frames describing the tag) from a number of movies. Table 1 shows our tag vocabulary which has 700 images pertaining to each tag. It is worth mentioning that some of the tags have overlapping features (e.g., violence, car chase, action, sword fight, etc) which makes this training problem tougher than the one in which classes share little to no features. Our tag vocabulary is scalable and evolving as we identify more relevant tags and collect the appropriate dataset. In order to construct the training dataset, we first formulate a criterion for finding the relevant training images pertaining to each tag. These images are collected as static frames from a number of movies. Table 1 also describes our semantic criteria of data collection for each tag which represents the required contents in an image describing a tag. From Table 1, it is also evident that we adequately cover the semantic contents pertaining to each tag by including its as many variants as possible.

A. TRAINING

The process of feature prediction with the pre-trained deep learning model and transfer learning with Softmax classification is depicted in Figure 1. We use Inception-V3 pre-trained model to retrain it on our tag vocabulary using transfer learning. After modifying the final classification layer of Inception-V3 model for movie tags prediction, we use the rest of the layers as fixed feature extractor. A dropout layer [54] is further added as a penultimate layer to randomly discard the activations of 50% neurons during training to prevent the inter-neuron dependencies and lack of generalization. A smaller learning rate of 0.005 with larger sizes (500) of training and validation batches are used to obtain more stable results. The dataset is partitioned such that 80% images are used for training, 10% for validation, and 10% for testing. For each input, the output of the penultimate layer, after applying dropout, is calculated as follows,

$$y_i = ReLU[\sum_j W_{i,j} x_j + b_i] \tag{1}$$

where $W_{i,j} \in \mathbb{R}$ is the weight coefficient associated with j^{th} and i^{th} neurons, and b_i represents the bias for i^{th} neuron. x_j represents the j^{th} activation of the feature map from the previous (convolutional) layer. Specifically, if this were the first hidden layer, x_j would be the j^{th} pixel of the input image. The Rectified Linear Unit ($ReLU$) activation function is used to restrain the output in a specific range. It is linear (identity) for all positive values, and zero for all negative values. The main purpose of an activation function is to introduce non-linearity and generalization in the training. Without an activation function, the CNN will be limited in its capacity to learn complex patterns and will behave akin to a linear regression model. The reasons of selecting ReLU activation function include its computational efficiency, smaller training time, faster convergence, and sparse activation.

The output of the penultimate layer is converted into a probability distribution. For this purpose, Softmax classification [2] is used to calculate the tag probabilities by the following rule,

$$p(i) = \frac{e^{y_i}}{\sum_j^{50} e^{y_j}} \tag{2}$$

where p_i is the probability of i^{th} tag in the tag vocabulary of 50 tags. p_i can be interpreted as the (normalized) probability assigned to the i^{th} tag. A cross-entropy error estimate [2] $E(p, q)$ is used to calculate the difference between the predicted distribution p and the actual distribution q by the following rule.

$$E(p, q) = -\sum_x^{50} q(x) \log p(x) \tag{3}$$

The cross-entropy $E(p, q)$ compares the model's prediction $p(x)$ with the label which is the true probability distribution $q(x)$. The cross-entropy decreases as the prediction gets more and more accurate.

The Softmax classifier aims to reduce the error estimate between the predicted and the true distribution. We train the model for 50,000 iterations (500 epochs). The description of the CNN training parameters is given in Table 2. The smoothed graphs of the training/validation accuracy and cross-entropy estimate are shown in Figure 2a and Figure 2b. It is apparent that the training-validation gap in both cases is

[1] http://www.image-net.org/

TABLE 1. Tags vocabulary and the semantic criteria used to collect data for each tag.

Tag	Semantic Description
Abduction	Scenes containing a forced escort
Action	Intense scenes, fight, rapid motion
Adventure	Scenes with exciting experiences, jungle, desert, etc.
Animal	Many types (birds, reptiles, mammals, insects, etc)
Animation	Scenes from cartoons, computer-designed movies
Beach/sea	Scenes showing sea shores or only sea with or without ships, boats, etc.
Bomb explosion	Scenes showing a bomb explosion and its destruction
Car chase	Scenes containing car pursuits
Children	Infants, kids
Climbing	Hill climbing, climbing on stairs, climbing with rope
Club/bar	Scenes showing a bar/club, people drinking in bar, people dancing in bar
College/university	Buildings of colleges/universities, libraries, students, academic activities, students in classes, etc.
Dance	Scenes in which people appear to move their body rhythmically, different dance moves
Desert	Movie scenes containing barren or sandy areas of landscape with little or no vegetation
Destruction	Scenes containing demolition and annihilation caused by natural calamities or human-induced disasters
Drama	Emotional or tragic scenes, people in dialogue
Drinking	People drinking wine/beer in traditional glasses
Exercise	Scenes showing physical exertion, practice, exercise, training
Family	Images of a family with parents, children and other members
Food	Images of a number of food types, edible items, people eating meals.
Forest	Scenes showing a large tract of land covered with trees and underbrush; woodland
Glamor/fashion	Scenes pertaining to modeling, fashion and demonstration of glamour, male/female fascinating poses
Heist	Movie scenes showing a robbery or holdup
Hiking	Scenes showing a walk or march through rural areas for pleasure, exercise, training or the like.
Horror	Scenes containing frightfully shocking, terrifying, or revolting contents
Hospital	Movie scenes showing hospital buildings, doctors, patients, wards, operation theaters, surgical procedures
Lab experiment	Scenes showing scientific experiments in laboratories
Military	Scenes showing people in military uniform and carrying weapons, artillery, tanks, army engaged in a war
Monster	Scenes showing grotesque and dreadful creatures having the forms of various animals in combination
Murder	Scenes showing killing or slaughtering inhumanly or barbarously
Music	Static frames of movies containing musical instruments, people singing on mic, concert
Nudity	Scenes containing partial or full display of both male and female bodies or genitals
Police	Movie scenes showing law-enforcement persons, police in action, police cars, police apprehending criminals
Prison	Images from the movies based on prison life showing prison buildings, cells, courtyards, inmates and their interaction
Robot	Images containing machines that resemble human and acting and responding in a mechanical, routine manner
Romance	Images of the intimate scenes from movies showing kissing or emotional attraction of a person towards another person
Science fiction	Scenes showing imagined future scientific advancements, space travel, life on other planets, etc
Sex	Scenes containing explicit sexual acts
Smoking	Images from the movie scenes showing people smoking or lighting cigarettes
Sports/athletics	Images of various types of sports, e.g., soccer, cricket, baseball, hockey, volleyball, car race, etc
Super hero	Images of heroic stock characters possessing supernatural or superhuman powers and dressed in a special costume
Swimming	Images of people swimming in a pond, river or sea
Sword fight	Scenes showing a war or a dual fought with swords
Technology	Scenes showing the use of advanced machinery, devices and electronic gadgets
Valleys/hills	Scenes showing hills or mountains with low areas of land between them
Vehicle crash	Images containing vehicles overturned or destroyed by an accident or head on collision
Violence	Scenes showing people scuffled, people using physical force intended to hurt or damage, bloodshed, gore contents
War	Scenes showing the armed conflict on a large scale (mostly involving military)
Weapon	Images of different devices used for attack or defense in combat, fighting, or war (e.g., gun, rifle, knife, etc)
Wedding	Scenes of wedding ceremonies with husband and wife wearing wedding dresses, wedding party, wedding celebration

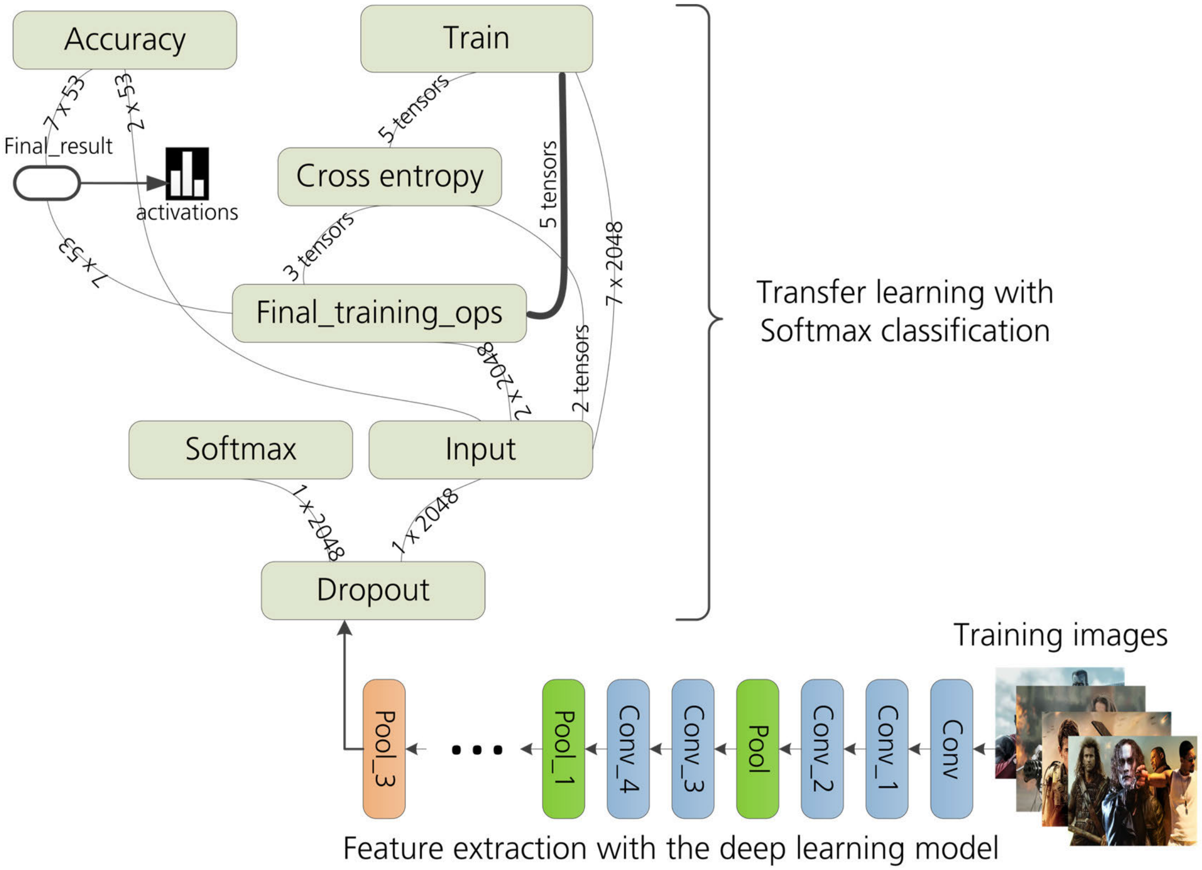


FIGURE 1. Feature extraction with the Inception-V3 CNN model and the subsequent training process using Softmax classification. In each training iteration, a training image is fed to the CNN model. The features extracted by the CNN layers are used for re-training the final classification layer.

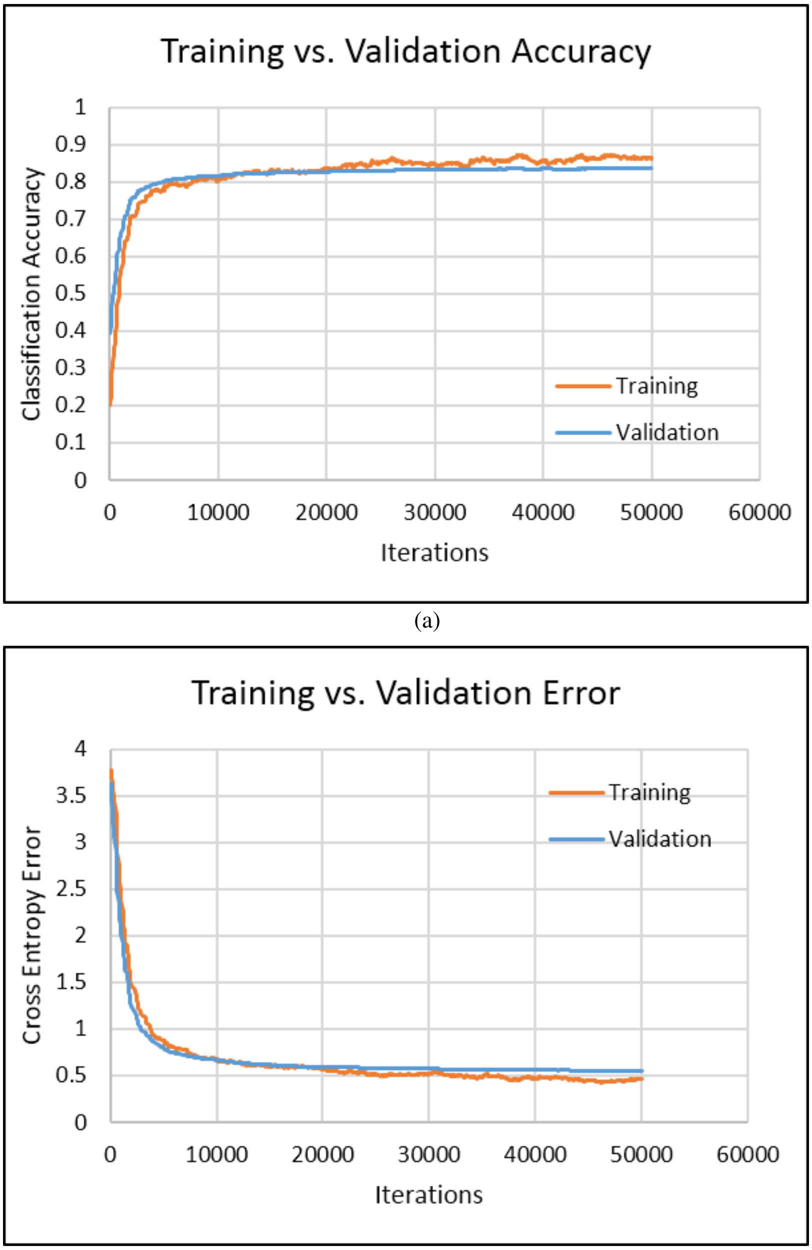
TABLE 2. Description of the parameters used to train the CNN model.

CNN Parameter	Description
CNN Model	Inception-V3, pre-trained on 1000 classes of ImageNet
No. of Layers	48
Types of layers	Convolution, AvgPool, MaxPool, Concat, Dropout, Fully Connected, Softmax
Modifications	- Final classification layer modified and re-trained - Addition of a dropout layer as a penultimate layer - Discarding the output of 50% neurons during training
Input image size	299 x 299
Learning algorithm	Softmax classification
Learning rate	0.005
Training & validation batch sizes	500
Dataset partitioning	80% training, 10% validation, 10% testing
Activation function	Rectified Linear Units (ReLU)
Error estimate	Cross entropy
No. of training iterations	50,000
No. of training epochs	500

significantly reduced. It is also evident that the addition of dropout layer and the right selection of training parameters leads to a good generalization. The overall test accuracy of the model is 85%.

B. TESTING

The trained model is tested on the static frames of different movies for tags prediction. The overlapping among the tag features further allows us to consider more than one predictions in the probability distribution. While the tag having the highest probability represents the most dominating content in a movie frame, the other tags lower in the probability distribution may also reveal important information. The results of tags prediction for some movie frames are shown in Table 3. The tags appear in the order of decreasing probability with the first tag representing the highest probability. Although



(a)

(b)

FIGURE 2. **Smoothed curves of (a) training-validation classification accuracy, and (b) training-validation classification error calculated by the cross-entropy function.**

the other tags have smaller probabilities in the probability distribution, yet they still reveal the relevant information contained in the movie frames.

C. SHOT BOUNDARY DETECTION AND KEY FRAMES EXTRACTION

For semantic analysis of a movie, it is inefficient to analyze all the frames in the movie. Instead, we first find the representative frames of all the shots in the movie. We find the shot boundaries and select the middle frame of each shot as the key frame. In order to find the shot boundaries, we compute the intersection of the HSV (Hue, Saturation, Value) histograms of the successive frames. This gives us a measure of similarity of the two discretized probability distributions (HSV histograms) with possible value of the intersection lying between 0 (no overlap) and 1 (identical distributions). The advantage of using HSV color space is that it is not only more robust to light variations, but is also better with respect to human perception [55].

Our shot boundary detection algorithm can detect two major types of shot boundaries: (i) hard-cut, which represents an abrupt transition from one shot to another, and (ii) dissolve,

TABLE 3. Predicted tags for static movie frames in the order of decreasing prediction probability.

Frames	Tags
	Military, action, weapon, war
	Violence, destruction, bomb explosion, action, vehicle crash
	Sex, nudity, romance, Glamor/fashion
	Hiking, adventure, forest, valleys/hills, climbing
	Sci-fi, super hero, robot, action
	Violence, sci-fi, action, horror

which is a gradual transition from one shot to another. We use a sliding window, centered on the current frame, on the similarity values of n frames. In order to determine hard-cut and dissolve shot boundaries, we use two adaptive thresholds which are based on the statistical properties of the sliding window. The adaptive threshold performs better than a single threshold which can not compensate for all the variations of the shot.

We evaluate the degree of similarity $S(i, i+1)$ between the current frame i and the next frame $(i+1)$ of the sliding window for the hard-cut boundary by first computing the intersection of the HSV based histograms for i and $(i+1)$. After it, the minimum m_1, second minimum m_2, and mean μ of the similarity values within the window are calculated. A hard-cut is detected between frames i and $(i+1)$ if the following three conditions are satisfied,

$$S(i, i+1) = m_1 \tag{4a}$$

$$S(i, i+1) \leq \alpha m_2 \tag{4b}$$

$$\mu \geq \alpha m_1 \tag{4c}$$

where $\alpha \in (1, 2)$. If the above conditions are not satisfied, we check for the dissolve boundary by the following rule,

$$\begin{aligned} S(i, i+1) &\leq \mu - (\mu - \sigma).\sigma \\ &= \mu(1-\sigma) + \sigma^2 \end{aligned} \tag{5}$$

where σ is the standard deviation of the similarity values within the window. It is worth mentioning that in comparison with the sliding window used to evaluate the hard-cut boundary, a window that remains fixed on the left side but grows on the right side by one frame after each evaluation, performs better for detecting a dissolve boundary. The algorithm of shot boundary and key frame detection is depicted in Algorithm 1.

Algorithm 1 Shot Boundaries and Key Frames Detection

```
Get $frame\_rate$ and $frame\_count$
Set $window\_size = frame\_rate$, $\alpha = 1.10$
for ($i = 1$ to $frame\_count$) do
    Convert frames $i$ and $i - 1$ to HSV channel
    Compute HSV histograms $h_1$ and $h_2$
    Calculate $hist[i] =$ intersect $(h_1, h_2)$
end for
Pad $(window\_size)/2$ null values in $window[\ ]$
for (all values in $hist[\ ]$) do
    Center $window[\ ]$ on $hist[\ ]$
    $mid = window[ceil(window\_size/2) - 1]$
    $window[\ ] =$ SORT($window[\ ]$,'ASCEND')
    $m_1 = window[0]$, $m_2 = window[1]$
    $\mu =$ mean($window[\ ]$), $\sigma =$ STD($window[\ ]$)
    if $mid = m_1$ and $mid \leq \alpha m_2$ and $\mu \geq \alpha m_1$ then
        $shot\_type =$ 'hard-cut'
    else if $mid \leq \mu(1 - \sigma) + \sigma^2$ then
        $shot\_type =$ 'dissolve'
    else
        $shot\_type =$ 'none'
    end if
    Get the shot boundaries and key frame
    Move the window to the next element in $hist[\ ]$
end for
```

D. TAGS PREDICTION

After finding a shot boundary and picking a key frame from the shot, we check if the key frame contains reasonable amount of information. For this purpose, we convert the key frame to luminance/chrominance color space and calculate the entropy of each channel by the following rule [56],

$$H(x) = -\sum_{i}^{n} \hat{p}(x_i) \log_2 \hat{p}(x_i) \tag{6}$$

where $\hat{p}(x_i)$ is the probability of a pixel x_i to have a certain value. Only those key frames are selected for semantic analysis whose cumulative entropy is greater than a certain threshold $(H > 0.20)$.

After finding the key frames of a movie, we feed each key frame to the trained model to obtain top 3 predictions. Subsequently, we determine the weight W_i of each tag by the following rule,

$$W_i = \frac{n_i}{N} \sum_{j=0}^{N} P_{ij} \tag{7}$$

where W_i denotes the weight of i^{th} tag, n_i is the number of occurrences of i^{th} tag, N is the total number of predicted tags, and P_{ij} is the probability of j^{th} occurrence of i^{th} tag.

The tag weights are further normalized in the range [0, 1] to calculate the relative strength R_i of each tag by the following rule,

$$R_i = \frac{W_i - W_{min}}{W_{max} - W_{min}} \tag{8}$$

where W_{max} and W_{min} represent the maximum and minimum tags strengths in the set of all the predicted tags. The tags having relative strengths less than a certain threshold are dropped to get a fewer key tags which best describe the movie. Figure 3 depicts the overall approach.

E. MOVIE SEGMENTATION

The shot boundary detection and analytics phase produces a corpus which contains the following information: (i) key frame of a shot, (ii) start and end frame of a shot, and (iii) top three predicted tags for each shot or its representative key frame. This information is used to merge the related shots based on a user-selected tag from the set of the predicted tags, as shown in Table 4. We select the extended trailers and long clips of several movies and first predict the set of key tags for each video. Using the movie analytics corpus generated during the process of tags prediction, we can segment a movie with respect to a selected tag based on top 1, top 2 or top 3 predictions for each key frame. We can also segment a movie based on more than one tags simultaneously (e.g., sex + nudity + romance, action + technology + sports, etc).

For example, Table 4 shows the analytic results of a movie whose set of tags are $t = \{$music, glamour/fashion, technology, family, college/university, club/bar, dance$\}$. After getting the tag set, we can segment the movie with respect to any tag. For example, if we segment the movie for finding only the scenes of music, the segmentation algorithm will first find all the shots related to music in the shot description table (Table 4). A shot is described by its boundaries, i.e., start and end frame. After finding all the shots corresponding to the tag of music, the shots are merged into a separate movie file which represents the segmentation of the movie with respect to the tag of music.

The accuracy of the extracted segment corresponding to a user-selected tag does depend on the number of top predictions (1, 2 or 3) selected to merge the related shots into a segment. For example, the tag of music can be the top 1, top 2 or top 3 prediction for a shot. If we select only the top 1 prediction for the segmentation of music tag, we will select only those shots in which the music tag is top 1 prediction (relatively smaller segment with higher precision). On the other hand, if we select top 3 predictions for the segmentation of music tag, we will select all those shots in which music is top 1, top 2 or top 3 prediction (relatively larger segment with smaller precision).

We find F1-score to be a better measure of the segmentation performance, since there is a relationship between the number of top predictions selected for the segmentation and the segmentation precision-recall. While the precision for small number of top predictions is higher with relatively lower recall, the opposite is true for higher number of top predictions selected for the segmentation. The results of this analysis are presented in Section V.

TABLE 5. Experimental setup.

Hardware/Software	Specifications
Microprocessor	Intel Xeon(R) E5430, 2.66GHz x 8
Random Access Memory	8GB
Graphical Processing Unit	GeForce GTX 1050 Ti, 768 cores, 4GB GDDR5
Deep Learning Framework	Tensorflow 1.2
Operating System	Ubuntu 16.04 (64-bit)
Programming languages	Python 3.0, OpenCV 3.0

TABLE 6. Evaluation results of the shot boundary detection algorithm for various movie trailers of varying lengths.

Trailer	Precision	Recall	F1-score
Grace unplugged (2013)	0.99	1.00	0.99
Frozen (2015)	0.97	1.00	0.99
Her (2013)	0.98	1.00	0.98
Inner workings (2016)	0.97	1.00	0.98
The silent child (2017)	1.00	1.00	1.00
Curfew (2012)	0.94	1.00	0.97
Dunkirk (2017)	0.94	1.00	0.97
God of love (2010)	1.00	1.00	1.00
Stutterer (2015)	1.00	1.00	1.00
The escape (2016)	0.97	1.00	0.98
Mean	**0.98**	**1.00**	**0.99**

V. EXPERIMENTAL SETUP AND RESULTS

Table 5 summarizes our experimental setup. We first find the shot boundaries and determine the key frame of each shot by the algorithm described in Algorithm 1. Figure 4a shows the detection of hard-cut shot boundaries which are represented by 'x'marks. These marks show the points where the conditions specified in equation 4 are satisfied. On the other hand, Figure 4b shows the detection of dissolve shot boundaries according to the conditions specified in Equation 5.

We first find the shot boundaries by watching all the movie trailers and noting down the start and end time of a shot. Using this information as a ground truth, we compare the shot boundaries with those determined by our shot detection algorithm. Table 6 shows the evaluation results of both hard-cut and dissolve shot boundary detection algorithm for various movie trailers of varying lengths. The size n of the window on the similarity values is taken as equal to the movie's frame rate (e.g., $n = 24$ for a movie having a frame rate of 24). It is evident that our shot boundary detection algorithm has impressive F1-score for both hard-cut and dissolve shot boundaries.

The average processing time for the whole algorithm, as shown in Figure 3, slightly varies with movie type. For the movies having more dynamic contents and consequently more number of key frames (e.g., action movies), the overall processing time is slightly higher. We evaluate the average time for the whole processing pipeline (including key frame

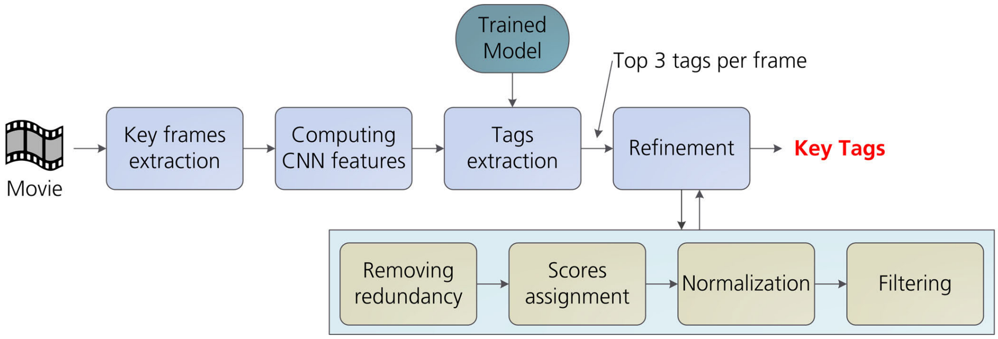


FIGURE 3. Overall technique of movie tags prediction. The fixed layers of the CNN extract features from a key frame and the final layer produces a probability distribution for all the tags. Top 3 tags for each key frame are selected for the refinement phase.

TABLE 4. The corpus generated by the tags prediction algorithm. For each key frame, top 3 tags are selected from the probability distribution.

Key frame	Timestamp	Shot Start Frame	Shot End Frame	Top 3 Predictions
⋮	⋮	⋮	⋮	⋮
853	0:0:36	832	874	Exercise Drama Glamour/Fashion
896	0:0:38	875	957	Glamour/Fashion Drama Exercise
971	0:0:41	958	984	Drama Romance Technology
1000	0:0:42	985	1016	Drama Wedding Glamour/Fashion
⋮	⋮	⋮	⋮	⋮

extraction, running inference on the key frames, and tags prediction) for 10 different movies. Using the experimental setup described in Table 5, the average processing time for a 720p resolution movie is 89 frames per second.

Since we do not have a ground truth for evaluating the performance of our tags prediction algorithm, we adopt a subjective criterion. Our subjective evaluation comprises 3 different experiments each performed on 10 different volunteers. For these experiments, we select a number of movie trailers of diverse categories. Not only a movie trailer best represents the whole movie, but it is also helpful to complete the experiments in reasonable time.

In the first experiment, the participants watched 50 movie trailers. At the end of each movie trailer, the set of its predicted tags were revealed to the audience and they were asked to judge its relevancy, accuracy and completeness by assigning it a score between 0 to 10 (0 being the worst and 10 being the best). A Mean Opinion Score (MOS) from the participants' feedback was calculated which is found to be 84.70%.

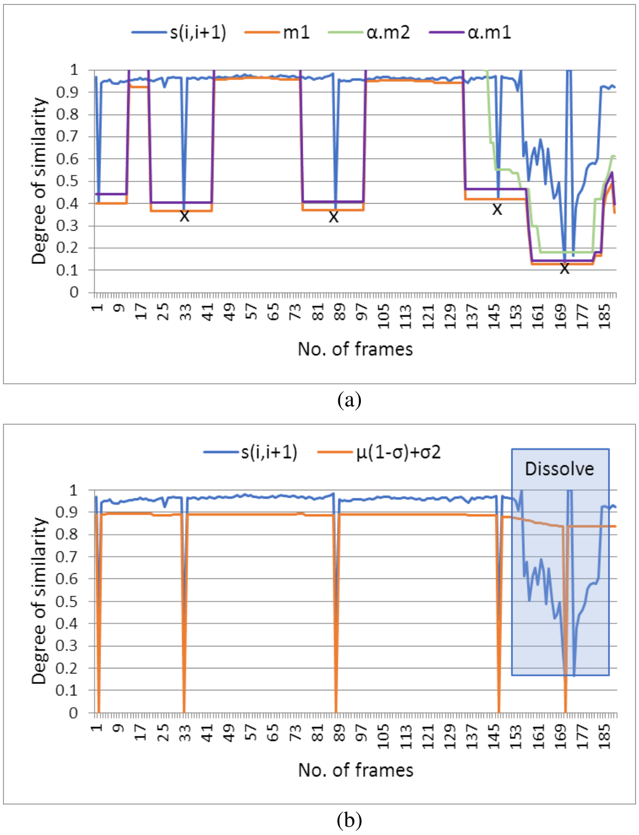
The second experiment was performed with different sets of participants and 50 different movie trailers. This experiment included asking the participants to rate the predicted tags after watching each movie trailer based on their relevancy as well as their relative strengths. This information was presented to them in the form of a visual chart as shown in Figure 5a. The MOS for this experiment was 79.20%. Figure 5b shows the predicted tags for the full length movie. It is evident that in both the cases, the sets of the predicted tags are similar with different relative strengths as there is a lot more information in the full length counterpart.

In the third experiment, performed with a different audience, the participants were handed over the whole tag vocabulary and were asked to watch a different set of 50 movie trailers. After watching each movie trailer, they were asked to point out appropriate tags for the movie trailer from the tag vocabulary. This experiment enabled us to calculate Mean Average Precision (MAP) P, Mean Average Recall (MAR) R, and F1-score by the following formulas,

$$P = \frac{1}{(MN)^2}\sum_{i}^{N}\sum_{j}^{M}\frac{t_p(i,j)}{t_p(i,j)+f_p(i,j)} \tag{9a}$$

$$R = \frac{1}{(MN)^2}\sum_{i}^{N}\sum_{j}^{M}\frac{t_p(i,j)}{t_p(i,j)+f_n(i,j)} \tag{9b}$$

$$F1 = 2(\frac{P \times R}{P+R}) \tag{9c}$$



(a)

(b)

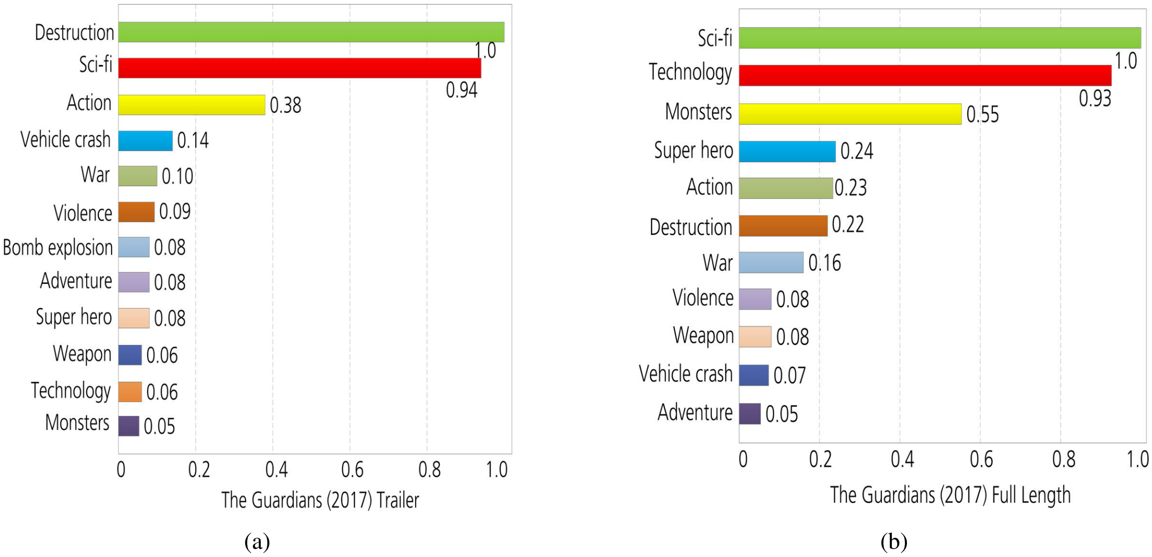
FIGURE 4. Detection of (a) hard-cut shot boundaries, and (b) dissolve shot boundaries. In (a), the hard-cut boundaries are represented by 'x' where the conditions specified in equation 4 are satisfied. In (b) the rectangle represents the dissolve shot boundary detected by the threshold in equation 5.

where $t_p(i, j)$, $f_p(i, j)$ and $f_n(i, j)$ represent the number of true positive, false positive and false negative, respectively, for the i^{th} movie trailer and j^{th} participant. Whereas, M and N represent the number of movie trailers and the number of participants, respectively. The MAP and MAR of this experiment were 76.50% and 74.55%, respectively, which gives a F1-score of 0.7551.

It is pertinent to mention that the manual annotation used for training does have the aforementioned limitations. It is because the background, experience, maturity, age, and qualification of the annotators can not be determined by random collection of data to construct a training dataset. However, in our case, the experiments have been designed in such a way that they not only involve the participants with known background, but also cover a wide variety of the movies to ensure the completeness of the experiments. In addition, as opposed to the traditional methods, our training does not rely on the manual annotation. It is only used for the evaluation. Hence, our subjective evaluation using three different types of experiments suffices to ascertain the efficacy of the proposed algorithm.

The evaluation of our movie segmentation technique with respect to three selected tags predicted by our algorithm is shown in Table 7. As discussed in Section IV-E, we can use $n = 1, 2, \ldots, m$ predictions per key frame for movie segmentation with respect to a selected tag, where m represents the total number of tags in the vocabulary. Since we pick only 3 topmost tags in the probability distribution of each prediction, the maximum value of m is 3. Hence, we can segment a movie with respect to 1, 2 or 3 topmost tags for each key frame.

In order to evaluate the efficacy of our segmentation approach, we first find the segmentation ground truth for a number of movie trailers with respect to top 3 tags. For this purpose, we carefully watch the movie trailer and find the shot boundaries (start and end frame) for each tag. We then compare the ground truth with the individual shots for each



(a)

(b)

FIGURE 5. Predicted tags of (a) a movie trailer, and (b) its full length movie. The relative weight of each tag is represented by the length of its bar.

TABLE 7. Evaluation of the movie segmentation technique with respect to the predicted tags. The results show a decrease precision and increasing recall by selecting top 1, 2 and 3 predicted tags for segmentation. The F1-score, however, increases.

			Top 1			Top 2			Top 3		
Movie trailer	**Main tag(s)**	**Segmentation tags**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Fast and Furious (2017)	Action	Car chase	0.95	0.75	0.84	0.84	0.88	0.84	0.75	1.00	0.86
		Vehicle crash	0.87	0.75	0.81	0.77	0.88	0.82	0.70	0.98	0.82
		Bomb explosion	1.00	0.62	0.77	0.86	0.70	0.77	0.81	0.81	0.81
Guardians of the Galaxy (2017)	Science Fiction	Monsters/Zombies	0.96	0.58	0.72	0.92	0.75	0.83	0.87	0.91	0.89
		Animal	0.97	0.50	0.66	0.89	0.55	0.68	0.89	0.57	0.69
		Animation	0.99	0.77	0.87	0.86	0.90	0.88	0.80	0.97	0.88
Wonder Woman (2017)	Adventure	Sword fight	0.80	0.90	0.85	0.80	0.99	0.88	0.80	1.00	0.89
		Destruction	0.89	0.50	0.64	0.78	0.68	0.73	0.70	0.79	0.74
		Adventure	0.91	0.69	0.78	0.81	0.87	0.84	0.76	0.96	0.85
Kill Command (2016)	Technology	Technology	0.94	0.86	0.90	0.88	0.94	0.91	0.87	1.00	0.93
		Science Fiction	0.76	0.64	0.69	0.72	0.75	0.73	0.66	0.85	0.74
		Military	1.00	0.60	0.75	0.96	0.80	0.87	0.90	0.90	0.90
Get on Up (2014)	Music	Dance+Music	0.92	0.84	0.88	0.90	0.93	0.91	0.90	0.95	0.92
		Club/Bar	0.98	0.86	0.92	0.98	0.92	0.95	0.95	1.00	0.97
		Drinking+Food	0.70	0.60	0.65	0.58	0.82	0.68	0.55	0.90	0.68
Chroniques Sexuelles D'une (2012)	Sex, Nudity	Sex,Nudity,Romance	0.98	0.84	0.90	0.97	0.95	0.96	0.96	0.98	0.97
		Family	0.82	0.60	0.69	0.80	0.91	0.85	0.76	1.00	0.86
		Drinking	0.93	0.70	0.80	0.83	0.91	0.87	0.81	1.00	0.90
Golden Shoes (2015)	Sports/Athletics	Sports/Athletics	1.00	0.94	0.97	1.00	0.96	0.98	0.96	1.00	0.98
		Children	1.00	0.70	0.82	0.95	0.85	0.90	0.90	0.96	0.93
		Family	1.00	0.70	0.82	0.75	0.82	0.78	0.70	1.00	0.82
Hecksaw Ridge (2016)	War/Military	Military	1.00	0.70	0.82	0.98	0.90	0.94	0.97	1.00	0.98
		War	1.00	0.50	0.67	1.00	0.72	0.84	0.97	1.00	0.98
		Weapon	1.00	0.50	0.67	0.80	0.60	0.69	0.69	1.00	0.82
The Void (2016)	Horror	Horror	1.00	0.85	0.92	0.95	0.90	0.92	0.90	0.98	0.94
		Monsters/Zombies	0.95	0.80	0.87	0.90	0.88	0.89	0.85	0.99	0.91
		Violence	1.00	0.70	0.82	0.98	0.90	0.94	0.97	1.00	0.98
Avengers: Infinity War (2018)	Super Hero	Super hero	0.96	0.82	0.88	0.92	0.90	0.91	0.90	0.96	0.93
		Action	0.90	0.78	0.84	0.88	0.86	0.87	0.86	0.92	0.89
		Science fiction	1.00	0.75	0.86	0.90	0.85	0.87	0.82	0.98	0.89
Mean			0.94	0.71	**0.80**	0.87	0.84	**0.85**	0.83	0.95	**0.88**

tag found by our segmentation technique. For each movie trailer, we compare the segmentation results with the ground truth for top 1, top 2 and top 3 predictions per key frame for each predicted tag.

Our evaluation shows an interesting relationship between the number of predictions per key frame used for segmentation and the precision-recall. Table 7 shows that as the number of predictions per key frame used for segmentation increases, the precision declines while recall increases. Nevertheless, this variation does not keep the F1-score same, as the change in recall is more abrupt than that in precision. Hence, the F1-scores for top 1, top 2 and top 3 predictions per key frame are 0.80, 0.85 and 0.88 respectively. While top 3 predictions per key frame give the highest F1-score, the trade-off between precision and recall further allows the user to segment a movie either with a higher precision (lower recall) or higher recall (lower precision) by selecting just one or higher number of predictions per key frame.

To the best of our knowledge, the relevant literature demonstrates no combined approach of movie tags prediction and the subsequent segmentation which can be used to compare the efficacy of our proposed technique. However, our detailed experiments suffice to demonstrate the performance of our proposed techniques.

VI. CONCLUSION

In this paper, we have proposed a movie tags prediction algorithm using deep learning. The predicted tags can be further used for segmenting a movie at the viewer's choice. Exploiting the powerful features of deep neural networks, we retrained a deep learning model (Inception-V3) using transfer learning to predict a class of a given movie frame from a carefully designed tag vocabulary. Subsequently, we proposed an efficient key frame detection algorithm which finds the representative frames of all the shots in a movie. Using the probability distribution of the prediction vectors generated by the final layer of the trained model for each key frame, we further proposed an algorithm which assigns weights to the predicted tags and finally produces a compact set of key tags which best describes the movie. The set of predicted tags can be further used to segment a movie using a corpus generated during the tags prediction algorithm.

Unlike the simple and limited approaches of movie tags prediction and segmentation studied separately or in combination in the literature, our proposed framework neither requires a priori knowledge of the tag features, nor is dependent on the user-annotated meta data which are major limitations of the techniques proposed in this context. In addition, our movie tags prediction and segmentation techniques are based on semantic analysis of the movie contents as opposed to the naive tagging and scene segmentation techniques studied in the literature.

We are also extending our tag vocabulary by identifying more classes and collecting the appropriate data. In future, we aim to extend our algorithm for audiovisual features. We believe that incorporation of audio features will further improve the performance of tags prediction for some classes (e.g., comedy, tragedy, etc) which are difficult to accurately predict using only visual features. In addition, we also aim to incorporate motion information in the prediction models by using recurrent neural networks which can better capture the dynamics of a scene by using a memory-based system.

REFERENCES

[1] U. A. Khan, N. Ejaz, M. A. Martinez-del-Amor, and H. Sparenberg, "Movies tags extraction using deep learning," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2017, pp. 1–6.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, vol. 1.

[3] P. Kim, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," Dec. 2015, *arXiv:1512.00567*. [Online]. Available: https://arxiv.org/abs/1512.00567

[6] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "C3D: Generic features for video analysis," *CoRR*, vol. 2, no. 7, Dec. 2014.

[7] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4694–4702.

[8] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, and G. Toderici, "Youtube-8m: A large-scale video classification benchmark," Sep. 2016, *arXiv:1609.08675*. [Online]. Available: https://arxiv.org/abs/1609.08675

[9] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, and H.-J. Zhang, "Correlative multi-label video annotation," in *Proc. 15th Int. Conf. Multimedia*, 2007, pp. 17–26.

[10] S. Siersdorfer, J. San Pedro, and M. Sanderson, "Automatic video tagging using content redundancy," in *Proc. 32nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2009, pp. 395–402.

[11] Z. Shen, S. Arslan Ay, S. H. Kim, and R. Zimmermann, "Automatic tag generation and ranking for sensor-rich outdoor videos," in *Proc. 19th ACM Int. Conf. Multimedia (MM)*, 2011, pp. 93–102.

[12] X. Liu, M. Corner, and P. Shenoy, "SEVA: Sensor-enhanced video annotation," in *Proc. 13th Annu. ACM Int. Conf. Multimedia*, 2005, pp. 618–627.

[13] J. Miranda-Steiner, "Automatic tag generation based on image content," U.S. Patent Appl. EP20 120 850 387, Nov. 16, 2016. [Online]. Available: https://www.google.com/patents/EP2780863A2?cl=en

[14] A. Ulges, C. Schulze, D. Keysers, and T. M. Breuel, "A system that learns to tag videos by watching youtube," in *Proc. Int. Conf. Comput. Vis. Syst.* Berlin, Germany: Springer, 2008, pp. 415–424.

[15] Z. Chen, J. Cao, Y. Song, J. Guo, Y. Zhang, and J. Li, "Context-oriented Web video tag recommendation," in *Proc. 19th Int. Conf. World Wide Web (WWW)*, 2010, pp. 1079–1080.

[16] W.-L. Zhao, X. Wu, and C.-W. Ngo, "On the annotation of Web videos by efficient near-duplicate search," *IEEE Trans. Multimedia*, vol. 12, no. 5, pp. 448–461, Aug. 2010.

[17] H. Aradhye, G. Toderici, and J. Yagnik, "Video2Text: Learning to annotate video content," in *Proc. IEEE Int. Conf. Data Mining Workshops*, Dec. 2009, pp. 144–151.

[18] G. Toderici, H. Aradhye, M. Pasca, L. Sbaiz, and J. Yagnik, "Finding meaning on YouTube: Tag recommendation and category discovery," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3447–3454.

[19] W. Yang and G. Toderici, "Discriminative tag learning on YouTube videos with latent sub-tags," in *Proc. CVPR*, Jun. 2011, pp. 3217–3224.

[20] W.-T. Chu, C.-J. Li, and Y.-K. Chou, "Tag suggestion and localization for Web videos by bipartite graph matching," in *Proc. 3rd ACM SIGMM Int. Workshop Social Media (WSM)*, 2011, pp. 35–40.

[21] S. Acharya, "Methods and systems for performing top concepts extraction," U.S. Patent 8 213 767, Jul. 3, 2012,

[22] T. Chen, C. Liu, and Q. Huang, "An effective multi-clue fusion approach for Web video topic detection," in *Proc. 20th ACM Int. Conf. Multimedia (MM)*, 2012, pp. 781–784.

[23] S. Kar, S. Maharjan, and T. Solorio, "Folksonomication: Predicting tags for movies from plot synopses using emotion flow encoded neural network," Aug. 2018, *arXiv:1808.04943*. [Online]. Available: https://arxiv.org/abs/1808.04943

[24] Q. Hoang, "Predicting movie genres based on plot summaries," Jan. 2018, *arXiv:1801.04813*. [Online]. Available: https://arxiv.org/abs/1801.04813

[25] M. Ullah, H. Ullah, N. Conci, and F. G. De Natale, "Crowd behavior identification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 1195–1199.

[26] H. Ullah, A. B. Altamimi, M. Uzair, and M. Ullah, "Anomalous entities detection and localization in pedestrian flows," *Neurocomputing*, vol. 290, pp. 74–86, May 2018.

[27] H. Ullah, M. Ullah, and M. Uzair, "A hybrid social influence model for pedestrian motion segmentation," *Neural Comput. Appl.*, vol. 31, no. 11, pp. 7317–7333, Nov. 2019, doi: 10.1007/s00521-018-3527-9.

[28] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 677–691, Apr. 2017.

[29] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," Dec. 2014, *arXiv:1412.4729*. [Online]. Available: https://arxiv.org/abs/1412.4729

[30] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence-video to text," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4534–4542.

[31] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, "Jointly modeling embedding and translation to bridge video and language," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4594–4602.

[32] A. Agrawal, J. Lu, S. Antol, M. Mitchell, C. L. Zitnick, D. Parikh, and D. Batra, "VQA: Visual question answering," *Int. J. Comput. Vis.*, vol. 123, no. 1, pp. 4–31, May 2017.

[33] M. Malinowski, M. Rohrbach, and M. Fritz, "Ask your neurons: A neural-based approach to answering questions about images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1–9.

[34] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.

[35] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov, "Exploiting image-trained CNN architectures for unconstrained video classification," Mar. 2015, *arXiv:1503.04144*. [Online]. Available: https://arxiv.org/abs/1503.04144

[36] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue, "Modeling spatial–temporal clues in a hybrid deep learning framework for video classification," in *Proc. 23rd ACM Int. Conf. Multimedia (MM)*, 2015, pp. 461–470.

[37] G. Boccignone, A. Chianese, V. Moscato, and A. Picariello, "Foveated shot detection for video segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 3, pp. 365–377, Mar. 2005.

[38] Z. Rasheed and M. Shah, "Scene detection in Hollywood movies and TV shows," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Dec. 2003, p. II–343.

[39] Z. Rasheed and M. Shah, "Detection and representation of scenes in videos," *IEEE Trans. Multimedia*, vol. 7, no. 6, pp. 1097–1105, Dec. 2005.

[40] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang, "Video summarization and scene detection by graph modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 2, pp. 296–305, Feb. 2005.

[41] Y. Zhao, T. Wang, P. Wang, W. Hu, Y. Du, Y. Zhang, and G. Xu, "Scene segmentation and categorization using ncuts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–7.

[42] Y. Zhai and M. Shah, "A general framework for temporal video scene segmentation," in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 1, 2005, pp. 1111–1116.

[43] Y. Zhai and M. Shah, "Video scene segmentation using Markov chain Monte Carlo," *IEEE Trans. Multimedia*, vol. 8, no. 4, pp. 686–697, Aug. 2006.

[44] V. T. Chasanis, A. C. Likas, and N. P. Galatsanos, “Scene detection in videos using shot clustering and sequence alignment,” *IEEE Trans. Multimedia*, vol. 11, no. 1, pp. 89–100, Dec. 2008.

[45] V. Chasanis, A. Kalogeratos, and A. Likas, “Movie segmentation into scenes and chapters using locally weighted bag of visual words,” in *Proc. ACM Int. Conf. Image Video Retr. (CIVR)*, 2009, p. 35.

[46] M. Hoai, Z.-Z. Lan, and F. De La Torre, “Joint segmentation and classification of human actions in video,” in *Proc. CVPR*, Jun. 2011, pp. 3265–3272.

[47] P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, H. Meinedo, M. Bugalho, and I. Trancoso, “Temporal video segmentation to scenes using high-level audiovisual features,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 8, pp. 1163–1177, Aug. 2011.

[48] M. Yeung, B.-L. Yeo, and B. Liu, “Segmentation of video by clustering and graph analysis,” *Comput. Vis. Image Understand.*, vol. 71, no. 1, pp. 94–109, Jul. 1998.

[49] H. Bredin, “Segmentation of TV shows into scenes using speaker diarization and speech recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2012, pp. 2377–2380.

[50] J. Baber, N. Afzulpurkar, and M. Bakhtyar, “Video segmentation into scenes using entropy and SURF,” in *Proc. 7th Int. Conf. Emerg. Technol.*, Sep. 2011, pp. 1–6.

[51] J. Baber, S. Satoh, N. Afzulpurkar, and C. Keatmanee, “Bag of visual words model for videos segmentation into scenes,” in *Proc. 5th Int. Conf. Internet Multimedia Comput. Service (ICIMCS)*, 2013, pp. 191–194.

[52] D. H. Nga and K. Yanai, “Automatic extraction of relevant video shots of specific actions exploiting Web data,” *Comput. Vis. Image Understand.*, vol. 118, pp. 2–15, Jan. 2014.

[53] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.

[54] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[55] J. Yue, Z. Li, L. Liu, and Z. Fu, “Content-based image retrieval using color and texture fused features,” *Math. Comput. Model.*, vol. 54, nos. 3–4, pp. 1121–1127, Aug. 2011.

[56] C. E. Shannon, “A mathematical theory of communication,” *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 5, no. 1, pp. 3–55, 2001.