Movie Description: An Interactive System for Visually Impaired People

Dr. Zulfiqar Ali Memon
Department of Computer Science,
FAST - National University of
Computer & Emerging Sciences,
Karachi, Pakistan.
zulfiqar.memon@nu.edu.pk

Luksh Kumar
Department of Computer Science,
FAST - National University of
Computer & Emerging Sciences,
Karachi, Pakistan.
K163642@nu.edu.pk

Ali Akbar
Department of Computer Science,
FAST - National University of
Computer & Emerging Science,
Karachi, Pakistan.
K163609@nu.edu.pk

Sateesh Kumar
Department of Computer Science,
FAST - National University of
Computer & Emerging Sciences,
Karachi, Pakistan.
K163910@nu.edu.pk

***Abstract*— This project is to automate the DVS (Dense Video Service), which is a manual method to write the descriptions of the clips of the movie for the visually impaired. This whole idea has been implemented using a sequential data processing pipeline starting from scene change detection to detect the timestamps in a movie where ever the environment or actions are changed drastically. These timestamps are given to the video trimmer module of the pipeline which is responsible to trim the movie into short video clips. These clips are then given to DVC (Dense Video Captioning) module, on which PCA (Principal Component Analysis) is applied to reduce the dimensionality, these features are then populated to LSTM (Long Short Term Memory) architecture responsible to generate a description of the clip. Face detection and recognition algorithm run over that same clip to detect and recognize the characters of the movie and then we substitute the character name into the description. These descriptions are then converted to audio format. These audios and video clips are then synchronized in such a manner that audio is played just before the video. We have been able to produce satisfactory results over a diverse range of movies and environments giving us a METEOR score of 9.65% as compared to the ground truth provided. This approach of movie description has the potential to be improved and scaled by working on the DVC module and generating more realistic and accurate descriptions.**

***Keywords*—Visual Impairment, DVS, Scene Change Detection, DVC, PCA, LSTM, ActivityNet, Face Detection, Meteor.**

I. Introduction

Visually impairment refers to the people that suffer from some form of vision malfunctioning or disorders, for instance, distance vision, near vision and complete blindness. According to the World Health Organization (WHO) study conducted in 2018, it is estimated that approximately 1.3 billion People live with some form of vision impairment out of 7.53 billion in total (17.26%), which makes them unable to visualize the environment and enjoy the beauty of nature and mankind. Therefore, there is a severe need from the computer science community to create efficient systems that can itself interpret the object, environment, and actions from visual data and transforms the insights into some other form like speech. More importantly, the media and entertainment industry play an important role in the social life of a person. But, being visually impaired, there are limitations to those experiences to which they are exposed, one of them is not being able to interpret and visualize the environment of movies. Thus, it highlights the significance of innovation and study required in this spectrum.

A product which will enable the visually impaired people to acquire the understanding of movies which in long term reduces the emotional, physical and societal gap between an ordinary human and a handicapped one. Suffering from such impairment causes the victims to be dependent on others to carry out their life tasks. Thus, this project enables its users to reduce the element of dependence from their life. Additionally, there is a national service called DVS which edit movies manually by describing key visual elements in scenes so that users can grasp the storyline of the movie. At first, specialists inspect through the movie for gestures and actions which can be missed by a blind person. They then transform those missing into a detailed audio so that blind people can engage with the movie in a better way. Unfortunately, this work requires educators, writers, trained musicians, and avid travelers to write such detail-oriented narration. Moreover, it is expensive and time-consuming process. Thus, after successful implementations, it will allow every impaired people to afford this luxury at their own convenience. Apart from that, the movie producers will be more engaged to generate the DVS of their movie with our automated service rather than paying heavily and investing a significant time for this.

One of the major applications of Video Captioning is in the area of describing the environment and actions of the movie that would eventually help the visually impaired people to visualize the movies in a better context and therefore, this highlights the dire need of an application that automates the movie description system. Furthermore, as stated in the literature review that no formalized application has been published till yet that performs the task of Movie Description in the full-fledged way. In addition, there are two major datasets based on movies data, however, what we have found in our research is that very few models have been implemented using the movies dataset (For instance, MVAD Dataset [1] and (MPII-MD) [2]) and majority of the work is done on the heavily recognized and well known datasets like MS-COCO and Flickr30 datasets which are used by majority of the researches worldwide to assess their model accuracy. As a result, many of the state-of-the-art CNN models like VGG16 and ResNet and RNN variants like LSTM have not been properly trained or evaluated on the datasets of movies that would have improved the accuracy of our Multi-Model to another level of heights. This eventually leads us to the conclusion that there is still some gap remaining that could further enhance the quality of descriptions and captions generated for each scene of the video by training the state of the art models on the movie dataset and then by comparative analyzing finding the best model for the best data and eventually implementing the best available technology for our target audience.

Another important lacking in the field of describing movies is the identifying, splitting and usage of the selective and specific scenes in a long video clip which should be used by our model to describe the overall scene to our user. This is

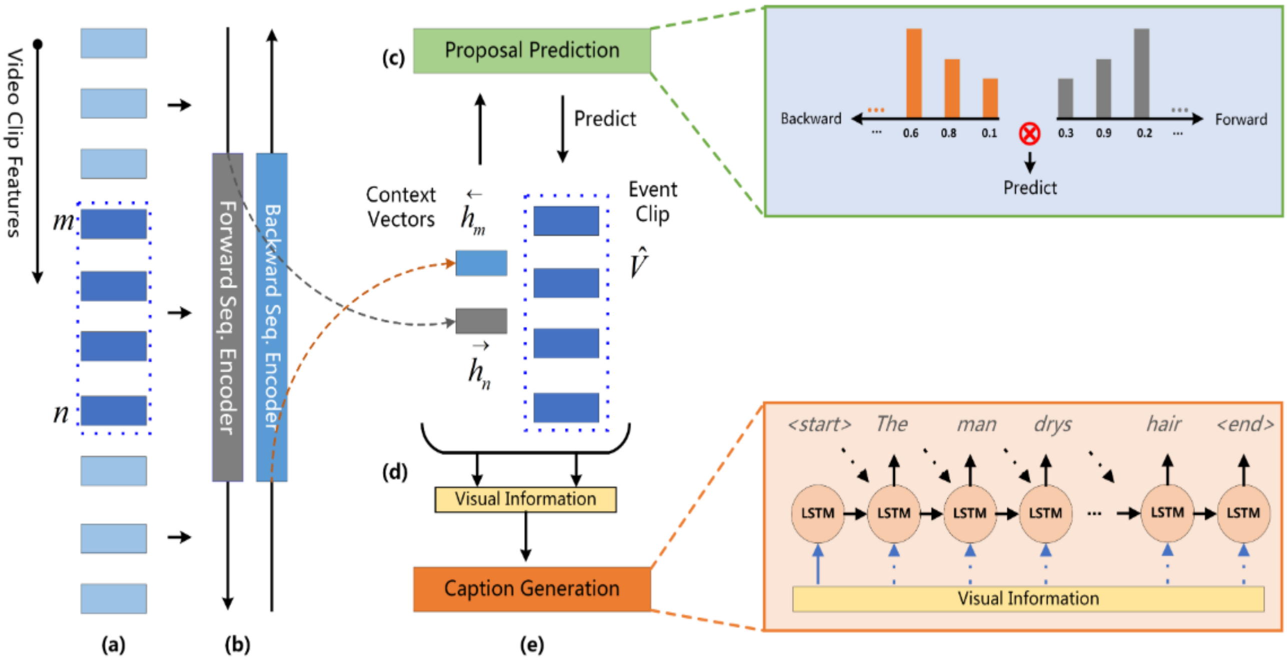

Fig 1. Fundamental Architecture for Dense Video Captioning

one of the major and prominent issues in the field of movie description because a movie is composed of several of the long snapshots which do not change for a longer period of time and hence, there should exist an intelligent model to detect this timeframe whose repetitive descriptions would not be beneficial for the visually impaired and may rather be a source of confusion to them. Therefore our proposed solution solves both of the above stated problems and therefore our approach is divided into four major parts: 1) Applying and training the state of the art models of CNN and LSTM on the movies dataset, 2) Evaluating the performances of the dataset on each of the models by using the test dataset from the DVS ground-truths and finally after critical analysis, selecting the best trained model, 3) Building the Machine Learning based algorithm that learns how to distinguish meaningful changes in the scenes ,and, 4) Incorporation of all models and algorithms and implementing and development of web based platform that integrates the trained model and synchronizes the movie audio with the audio description generated by our movie. In summary, the final product would be a web application would take a movie (video) as an input, process the video and generate the captions, converts the textual description to audio, and finally plays the movie by first identifying the segments of the video of which to describe and then synchronizing the audio of the movie and the generated audio description.

II. Literature Review

A. Image Captioning

Automatic textual description of the image is studied enormously in the past including [3, 4] which entirely focus on descripting the images using object detection and action recognition from computer vision literature. The majority of research has been conducted on Convolutional neural network (CNN) architecture and Recurrent Neural Network (RNNs) or specifically Long Short-Term Memory (LSTMs) for the textual generation. There has been exponential growth in this area after the evolution of CNN, which leads to the formation of many dense datasets for model evaluations such as MS COCO [5] and Flickr 30 [6] which are still the state of the art for image annotation tasks.

B. Video Captioning

The idea of image captioning leads to the formation of video captioning which is not limited to the object detection and action analysis, rather it formulates the textual description from a series of frames used to form a video. The majority of initial work gives a single line explanation of the whole scene except [7] which focus on multiple sentence video descriptions with variable length of the text.

Another similar approach has been discussed thoroughly in [8] explaining the text using temporal structure of the video frames. These methods were also based on CNN features with the max pooling including the LSTM as recurrent network.

C. Descriptive Video Service

Descriptive Video Service (DVS) is a national service that makes television programs, feature films, home videos, and other visual media accessible to people who are blind or visually impaired by manual annotations making textual descriptions of the movie scenes including the actions and its environment.

This is a very costly process since it requires artist and movie writer to sit and analyze the movie to write its descriptions. This textual description was later on converted into audio by getting it voice over from other resource. This work has been tried to make it automatic by stretching the idea of video description to movie scenes. There are many different researches works on the dataset creation for the movies such as [9] which has tried to make the dataset from segregating the dialogues from the narration using the semi-automated processes. There are some audio descriptions available for the movies which has been done manually for different movies and that is basically used to create the datasets for the movies.

Once the textual description is formed, describers watch the program and write a script describing key visual elements. They carefully time the placement and length of the description to fit within natural pauses in the dialogue. After a script is completed, it is edited by a post-production supervisor for continuity, clarity, and style conventions. Narration is recorded and mixed with the original program audio in a unique "mix to pix" process to create a full DVS track. This is how Media Access Group formulate this process which we aim to automate.

D. *Movie Description*

The two recent movie datasets that have been published are Montreal Video Annotation Dataset (M-VAD) [1] and MPII Movie Description (MPII-MD) [2] which covers a broader domain compared to the initial video descriptions. They are now being recently in many of the research work and we aim to use the same for this project. The till date work on the movie description is so trivial that cannot be generalized to deploy and make a product out of it which can let the audience input the movie and generate its textual description and convert it to speech format, enabling the blind to actually use this research in daily life. This similar sort of work has been done on short video clips and images for the research purpose but there is no such product that we convert the movie scenes information from one medium into another medium on a fly.

E. *Existing Approaches*

This whole manual process is very time taking and expensive that is the reason we do not have audio description for each and every movie. This makes a great opportunity for the researches to make this process automatic in order to convert it into a financial product that can itself turn out to be a business aid. Therefore, using the existing literature we could found out that automatic textual description of the image is studied enormously in the past including [2, 3] which entirely focus on descripting the images using object detection and action recognition from computer vision literature.

The idea of image captioning leads to the formation of video captioning which is not limited to the object detection and action analysis, rather it formulates the textual description from a series of frames used to form a video. The majority of initial work gives a single line explanation of the whole scene except [6] which focus on multiple sentence video descriptions with variable length of the text. Another similar approach has been discussed thoroughly explaining the text using temporal structure of the video frames. These methods were also based on CNN features with the max pooling including the LSTM as recurrent network. Despite of having significant literature on this problem, we still cannot have a business solution towards this problem because the whole pipeline of converting movies into audio format and synchronizing it is not trivial, therefore this is what the real need and gap in the market and research industry has been suffering and is one of the reason why audio descriptions are yet not automated.

III. METHODOLOGY

A. *Scene Change Detection*

The project entitles many different but correlated modules incorporating their theoretical details which is covered in depth in this section. This pipeline of producing highly dense descriptions for a movie begins with the most crucial discuss of theoretical aspect of scene change detection. The idea is to detect the timestamp where ever a scene i.e. the background or actions changes drastically. This has been significant literature on this work mainly including two widely used approaches for scene change detection incorporating pixel-based and histogram-based methodologies.

The technique of comparing the intensity values of two consecutive frames of a video is what the idea of pixel-based scene change detection. We set a threshold for the minimal difference between the two consecutive frames via which we can filter out whether that difference is because of some camera movement or illumination changes or is there a significant background change from indoor environment to outdoor environment or a major change in the actions of the characters. This approach has been densely used for years but the hyperparameter is the key factor here which is very difficult and cumbersome to decide because this peculiarity varies from application to application. Therefore, there are significant problems in using pixel-based approaches because it does capture those scenes which are not drastically different than that of previous scenes. This is because we do have some brightness or illumination changes as far as the application of movies are concerned. The below illustrates how a scene change with pixel-based approach would take out the different and set the threshold on it to classify it.

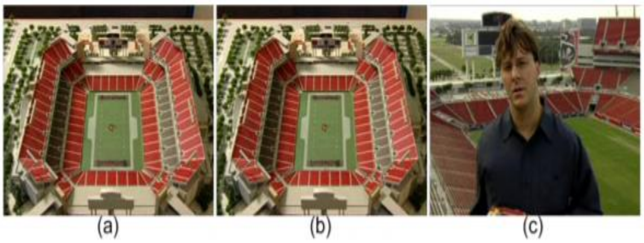

Fig. 2. The pixel-based would recognize the image a and b as same because the difference between the intensity values for both images are almost same. Whereas, the difference between the image b and image c is drastic because of the man in the foreground. These significant intensity changes would cross the threshold and would be classified as a scene change.

The second widely used approach is to calculate the distribution of the two consecutive frames which can be visually analyzed by plotting a histogram and comparing the bins frequency for each frame. If the two consecutive frames are drastically different, then their distribution across the bins of the histogram would be different as well. This would reflect that there is a significant background or action change. We also require a hyperparameter of threshold here which helps to classify whether the certain distribution is drastic enough to be classified positively or not. This approach is far better because it compensates for the camera movement and illumination problems but it does suffer from problems where there is a scene change but coincidently the distribution of the frames remains the same. This happens when the scenes are different but with that small region and not entire in the whole picture. Therefore, this tradeoff between the pixel-based approach and histogram-based approach is well covered in [9] which focuses to merge both of the approaches in a certain way by setting another hyperparameter which decides the weightage of each approach. In this application of movies, we tend to have more gradual changes rather than abrupt and frequent changes we see in general YouTube videos. After calculating the distributions, a line is plotted having the bins frequencies plotted on x-axis for one frame and on y-axis for another.

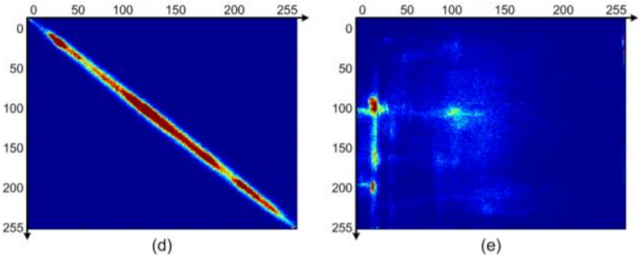

Fig. 3. The histogram-based approaches plot these line graphs to detect a change. If there is a straight diagonal line, that implies a stagnancy and no scene change, while an unorthodox curve shows that two frames are entire difference implying a scene change. As we can compare from Fig 2, the image d line is of image a and image b, while image e plot is for image b and image c from Fig 2.

B. Dense Video Captioning

We used an approach as mentioned in [10] to densely generate video descriptions. The paper introduces an improved version of a well-known approach known as Single Stream Temporal Action Proposals (SST), previously mentioned in [11] for video action proposal generation which requires the localization of all possible events that occur in a video. The improved version provides a Bidirectional version of SST which allows the network to efficiently encode both past, current, and future video information. This methodology is further illustrated in Fig 1, where the video is first encoded as a sequence of visual features using the C3D Network which is trained on Sports-1M video dataset. Then, PCA is applied on these features to reduce the dimensions of the 4096-feature vector to 500 most valuable components. The reduced features are then passed to the bidirectional SST during which the forward pass encodes past context and current event information, while the backward pass encodes future context and current event information. Finally, the proposals with the same scores for predictions are merged and the final proposals are presented. Thereafter, after fusing the proposal state information and the C3D sequences, these representations of proposals are passed to the captioning module which is a network of LSTMs to generate language descriptions. To run this module, we used the model that was trained on ActivityNet Captions dataset which contains about 20,000 videos (849 video hours) and about 100,000 total descriptions, where each sentence has an average length of 13.48 words.

This module is responsible to detect the faces and recognize the characters within those such that the general names like man, person, woman can be replaced by the actual character names.

C. Facial Recognition

After applying the process of DVC, we have the description based of each video clip, however, since our project describes the visual information in a movie for visually impaired people, it is necessary to replace the general tags of people in the descriptions which includes tags such as “man”, “woman”, “someone”, “girl” and “boy” into the actual character names appearing in the video so that the scenario much more evident for the user. To achieve this, the first step in our approach was to find all the faces in the video frames, therefore we used a technique known as Histogram of Oriented Gradients (HOG), as shown in Fig 4 to detect all the faces in a given frame as proposed in [12]. So, at this point, we isolated the faces in the frames, however, since different pictures of the same person may look different in representation to the computer, we used a technique called Face Landmark Estimation, proposed in [13]. Based on these landmarks and by using Machine Learning models, we were able to retrieve 68 specific points on a face so that for a distorted face, a set of transformations can be applied to get the eyes and mouth centered as best as possible. We then used a Deep Convolutional Neural Network to produce a 128-dimensional encodings vector for any given face. Finally, we provided the images for all the actors for a given movie and wrote a program that computed the encodings for the given set of known faces and compared it versus the faces seen in the running video. If the encodings match up to a certain threshold, the specific character name is written onto a file based on the sequences of occurrences.

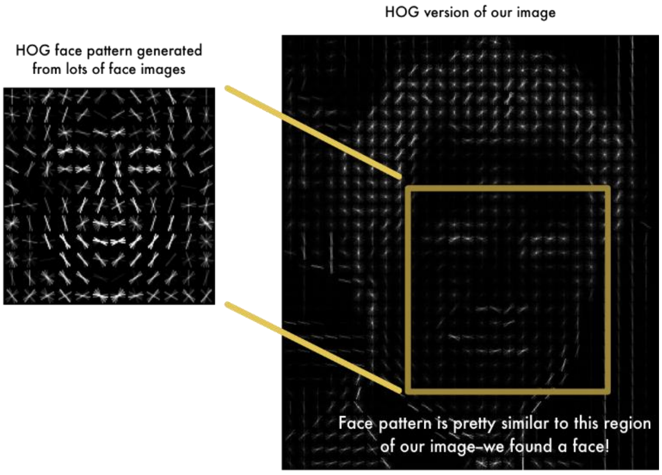
HOG version of our image
HOG face pattern generated from lots of face images
Face pattern is pretty similar to this region of our image–we found a face!

Fig 4. The usage of HOG method to extract face patterns for face detection.

D. Replacement of General Tags with Character Names

Till now, we have the description and the character names that occurred in the video and all we have to do is replace those tags with the character names based on occurrences. There are several approaches that can be used to achieve this behavior, one of which is mentioned in [14]. We have also used a similar approach which given a video track looks out for the general tags in the description for the specific track and replaces those tags with the names of the characters that appeared in the movie sequentially. A more generalized and effective technique is discussed in [14] that improvised of first getting actions from a video clip and then creating a Neural Network based Textual-Visual Embedding Space which projects input tracks and verbs and then based on character names from facial recognition module, it maps the verbs to faces and based on this association and distance metrics , it replaces the generic tags with the names. However, the stated process needs a specialized cleaned and well processed data especially for this purpose and in addition requires huge amounts of training which was unavailable and under evolution at the time of writing this report. Therefore, a well suited and much effective technique as stated has been deployed to achieve the required results.

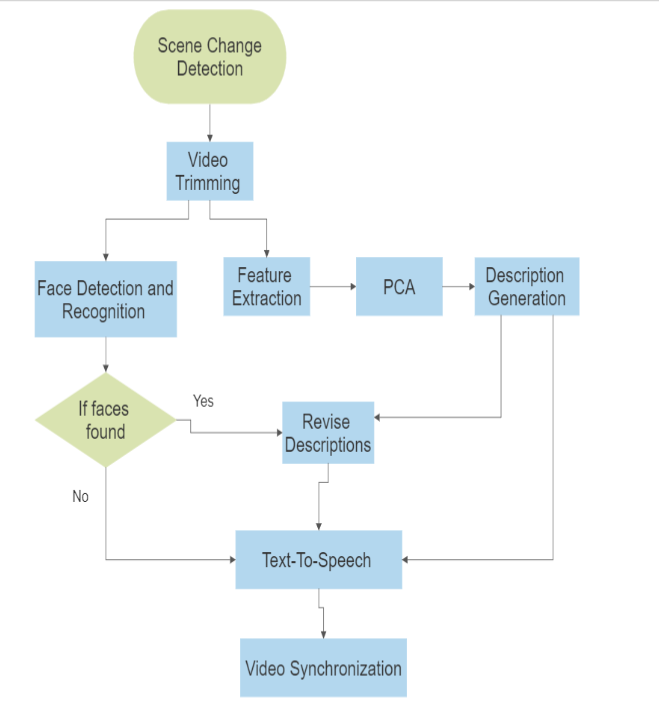

Fig. 5. Flow-Chart representation of the entire data processing pipeline. The input video is first processed through scene change detection module, passed to video trimming for dividing video into clips, then there is a parallel process of face detection and recognition on those clips along with feature extraction and description generation. If faces are found, they are replaced in description else the same description is passed to Text-To-Speech for audio conversion and finally those clips and audio file is combined by synchronizer to output a processed movie.

IV. Data & Results

The project is completed utilizing two major datasets involving Sports-1M trained on sports videos and ActivityNet trained on general YouTube videos. The purpose of using Sports-1M dataset is to get 4096 feature videos given a video. The architecture trained on this dataset was 3D Convolutional Neural Network which is renowned for human activity recognition because of working on volume of image and filters rather than working on single 2D image and kernels. This benefits in capturing the past information of the frames, resulting in robust feature extraction capturing all of the actions. These features are then passed through a dimensionality reduction algorithm known as principal component analysis which gets 4096 as the input space and returns 500 most valuable and informative principal components. These components are then given to LSTM trained which is trained on ActivityNet dataset incorporating general videos. The rationale behind using ActivityNet for dense video captioning is because it is trained on highly dense description which are based on grammar used for compound sentences forming a paragraph rather than just giving a single sentence as output.

The product's output is shown and described in detail but with respect to the data processing pipeline as described in Fig 5.

A. *Scene Change Detection*

The scene change detection gives the output in form of the visual snaps and the timestamps at the top of it, we gather these timestamps and then consolidate into one file for processing as shown in Fig 7.

B. *Video Trimming*

Once the timestamps are generated from the scene change the video trimmer cuts the whole movie into small chunks or clips that looks like these. The movie named 3.mp4 is clipped into four different chunks with respect to scene change.

C. *Dense Video Captioning*

Once the feature extractor is run, PCA is applied, we get the textual description in the last for all the clips as shown below. These descriptions are then saved as the text file.

D. *Face Detection and Recognition*

Once we get the description, we run the face detection and recognition module to replace the generic names with characters names as demonstrated in Fig 6 and 8.

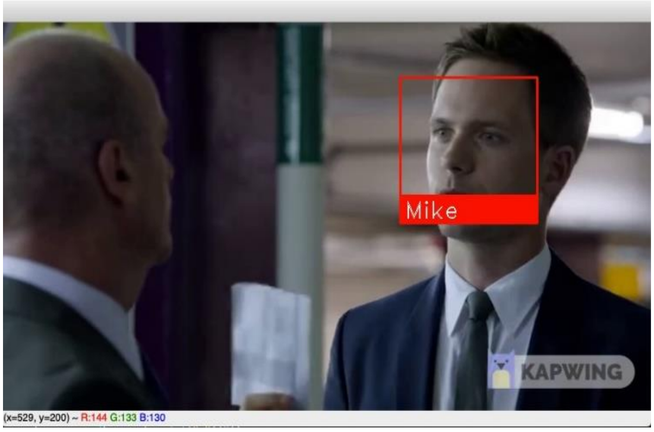

Fig. 6. The Face Detection and Recognition of Mike in first clip of "Suits – English Sitcom".

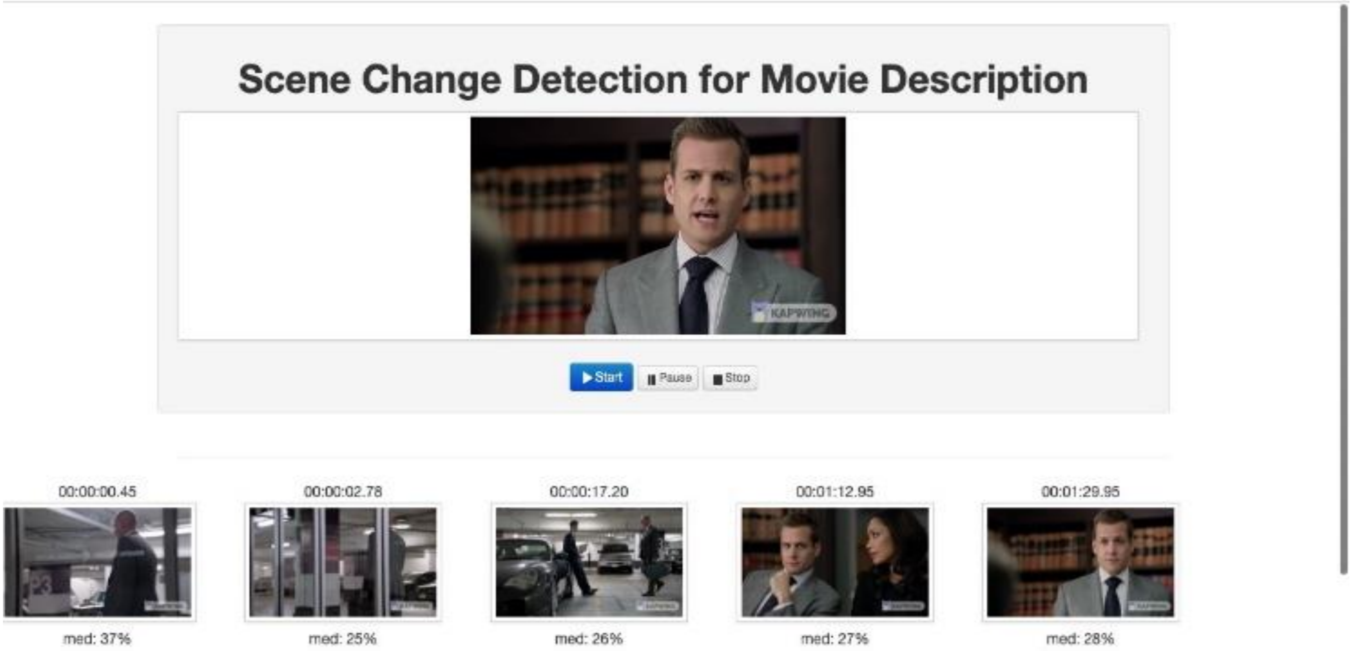

Fig 7. Scene Change Detection for one of the Clip of the "Suits – English Sitcom".

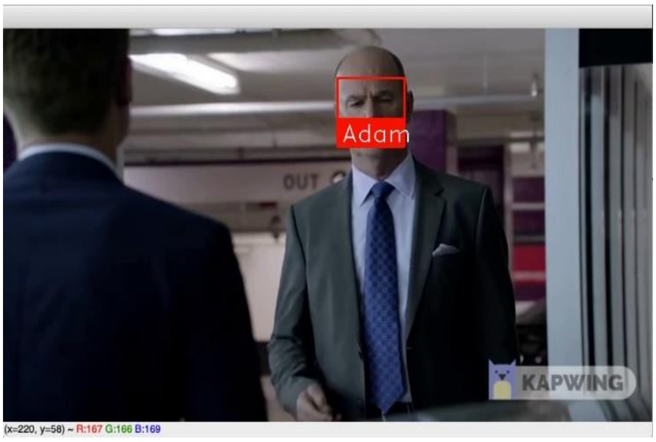

Fig. 8. The Face Detection and Recognition of Adam in first clip of "Suits – English Sitcom".

The finalized description after replacement of generic tags with character names for this particular clip is as following.

LSTM: Mike is seen speaking to the camera and leads into Adams pulling up a piece of exercise equipment. Mike is seen speaking to the camera while holding up a piece of paper.

E. *Text-To-Speech*

Once the finalized description is generated, we run the API to generate audio files for the descriptions so that the blind person can listen to it. These files are generated with the same name as their clip.

F. *Video Synchronization*

The finalized and processed movie is then ready on the webpage for the user to watch. As soon as the pipeline is completed processing, it is automatically reflected back on the website without a page reload. In addition to that, a notification can be generated for longer movies so that a user can be informed only when the entire pipeline has finished processing.

G. *Quantitative Results*

This appraoch of dense video captioning with bi-directional single stream temporal enabling the sliding window to look forward as well as keeping the information retained for all the consecutive sequences that has been captured before helps to find more robust and accurate temporal event proposals. These proposal are then feeded to LSTM for their separate descriptions building the foundation of dense video captioning. Therefore, this bi-directional technique has led the model to accurately describe the situation resulting in METEOR score of 9.65 as compared to the 4.82 of Krishna *et al* [15].

Method	**Meteor**
Krishna *et al*	4.82
Bi-SST	9.65

V. Future Work

This project has a diverse range of improvement opportunities and research-oriented areas where the specific in-depth research would help to improve the total outcome of the project. The most critical and important part of the pipeline discussed in this report and implemented in this project is the dense video description in which we convert the visual information into textual format, this area using the deep learning architecture needs a lot of improvement in terms of more data, more intensive training on the movies datasets rather than building the architectures on general purpose videos because movies have different rate of change of environment and actions which are usually pretty slow and less abrupt. Moreover, this work can be extended by working on scene change detection process in which a more robust approach would allow the scene change to work better on fast forward mode enabling the user to get output i.e. the processed movie less than the total duration of the movie. In this work, the scene change is working well in play mode but not with fast forward mode. In addition, a better hardware availability and resources would allow the upcoming researchers to test the data on higher batch size which would allow the model to give most accurate outcomes, since in our case we were running out of memory whenever we tried to test the samples above 50 batch size of about 1 – 1.5 minutes of movie at 25 frames per second per second.

VI. Conclusion

This project provides an automatic mechanism of audio description i.e. converting the environment and actions of any movie into audio format and synchronizing it with the movie so that it becomes possible for the visually impaired people to interpret and imagine the scene with the help of movie dialogues and background voice. This enables many producers to get their movies available for diverse range of people because of its low cost and time saving automatic audio description generation. This work included a sequential pipeline of 6 modules including scene change detection, video trimming, dense video captioning, face detection and recognition, text-to-speech and finally the video synchronization process. This project has been given a shape of business oriented financial product by building a web portal where a user can interactively get their movie converted with ease and can also watch it using the video player available on the site.

REFERENCES

[1] Torabi, A., Pal, C., Larochelle, H., Courville, A.: Using descriptive video services to create a large data source for video annotation research. arXiv:1503.01070v1 (2015)

[2] Rohrbach, A., Rohrbach, M., Tandon, N., Schiele, B.: A dataset for movie description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)

[3] Farhadi, A., Hejrati, M., Sadeghi, M., Young, P., Rashtchian, C., Hockenmaier, J., Forsyth, D: Every picture tells a story: Generating sentences from images. In: Proceedings of the European Conference on Computer Vision (ECCV) (2015)

[4] Mitchell, M., Dodge, J., Goyal, A., Yamaguchi, K., Stratos, K., Han, X., Mensch, A., Berg, A.C., Berg, T.L., III, H.D.: Midge: Generating image descriptions from computer vision detections. In: Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL) (2012)

[5] Chen, X., Fang, H., Lin, T., Vedantam, R., Gupta, S., Dollr, P., Zitnick, C.L.: Microsoft coco captions: Data collection and evaluation server. arXiv:1504.00325 (2015)

[6] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, Svetlana Lazebnik: Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models (2015)

[7] Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Transactions of the Association for Computational Linguistics (TACL) 2, 67–78 (2014)

[8] Rohrbach, A., Rohrbach, M., Qiu, W., Friedrich, A., Pinkal, M., Schiele, B.: Coherent multisentence video description with variable level of detail. In: Proceedings of the German Confeence on Pattern Recognition (GCPR) (2014)

[9] Sung In Cho, Suk-Ju-Kang: Histogram Shape-based Scene-Change Detection Algorithm, IEEE (2019)

[10] Wang, Jingwen and Jiang, Wenhao and Ma, Lin and Liu, Wei and Xu, Yong: Bidirectional Attentive Fusion with Context Gating for Dense Video Captioning (CVPR) (2018)

[11] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. Carlos Niebles: SST: Single-stream temporal action proposals. (CVPR) (2017)

[12] Deniz, Oscar & Bueno, Gloria & Salido, Jesús & De la Torre, Fernando. Face recognition using Histograms of Oriented Gradients. (2011)

[13] Vahid Kazemi and Josephine Sullivan KTH: One Millisecond Face Alignment with an Ensemble of Regression Trees (CVPR) (2014)

[14] S. Pini, M. Cornia, F. Bolelli, L. Baraldi, and R. Cucchiara: M-VAD Names: a Dataset for Video Captioning with Naming (2019)

[15] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei : Dense Video Captioning in Videos arXiv:1705.00754v1 (CVPR) (2017)