Text Detection And Recognition From The Captions Of Streaming Videos Using Tracking

Sasanko Sekhar Gantayat, Soujanya Yellumahanti

Abstract: Text is a part of direct source of information in a scene image. But it is a great challenge for us to detect, extract and recognise the text by seeing a scene image due to the difference in size, style, orientation, robustness to background complexity, text degradation and distortion, text variations, and moving objects. A few applications are helps for outwardly impeded people, interpreters for sightseers, data recovery frameworks indoor and outdoor environment and programmed robot route. Existing strategies for scene content recognition can be arranged into three groups: sliding window-based, associated segment-based AdaBoost Classifier and Clustering systems. Sliding window based strategies are otherwise called region-based techniques; the sliding window is utilised to scan for potential texts in the image and after that utilisation AI methods to recognise text. Connected component-based methods separate CCs from images by connecting-components by grouping character-candidates into the text; extra checks might be performed to dispose of the non-text regions. The AdaBoost technique is for connecting the clusters by their pairwise relations. The AdaBoost technique is for connecting the clusters by their pairwise relations. MSER algorithm is used to check the connected clusters are connected correctly or not. Analysis of video data is of high prominence nowadays and text in videos is a chief source of information in them. Extracting text from videos has been implemented in many ways. It is focused on tracking text with detection and recognition of text. Using on a Bayesian framework with bilateral filter, we carry out the extraction of embedded caption text in web videos. The Bayesian framework of tracking based text detection and Recognition(T2DAR) from videos is used, which has significant tasks of text tracking. The missing detection is retrived by tracking the text is tuned the scales of detected tracking based text recognition. An agglomerative Hierarchical Clustering is implemented to cluster the over segmented trajectories. This approach of tracking the text and detection has given a better result when compared to existing ones.

Index Terms: MSER Algorithm, Bayesian-based Framework, Bilateral Filter, Text Extraction, OCR.

1 INTRODUCTION

The present world is totally computerised and headways are improving step by step. With extending advances, the necessity for computerised at this point effective frameworks is expanding. The development of advanced gadgets like PDAs, workstations, tablets brought about a collection of much visual information, particularly recordings on the Internet. These have activated research exercises in getting interactive media and enthusiasm for separating substance in the video, including significant data and printed information. Text Detection, Recognition and Tracking has risen as a significant issue in computer vision community for the last few years because several research activities include mobile devices which are furnished with high-resolution cameras in for human-computer interaction (HCI) and received more attention and text is also easily recognized by machines which are useful for various applications. There are many workshops and conferences like the International Conference on Document Analysis and Recognition (ICDAR) are being organized for further developments in text processing from the image documents. The optimization of multimedia database access and scheduling is done by the image retrieval process. Video recognition is based on the video access and retrieval to deliver accurate indexing and explanation.

- *Sasanko Sekhar Gantayat, Department of Computer Science and Engineering, GMR Institute of Technology, Rajam, Andhra Pradesh, India, E-mail:sasankosekhar.g@gmrit.edu.in*
- *Soujanya Yellumahanti, Department of Computer Science and Engineering, GMR Institute of Technology, Rajam, Andhra Pradesh, India, E-mail: yellumahantisoujanya@gmail.com*

The traditional method of video text recognition includes manual explanation, text detection, encoding the image and feature extraction. The method has large interfering in the overlaid text detection, particularly the automatic recognition ability of the overlaid text recognition and retrieval has low accuracy. In this paper, a conventional Bayesian structure of T^2DAR for installing proposed subtitle content performs both follows the content identification and content acknowledgment in a solitary brought together pipeline

When all is said in done, the input data between the identification and following acknowledgment in complex recordings are trying for misusing and sharing. In this work, a brought together a plan of both following based content identification and following based content acknowledgment is structured inside a Bayesian system. To characterise, Tracking is the undertaking of keeping up the uprightness of the content area and the following content crosswise over adjoining outlines. Location is the undertaking of limiting the content in every video outline with jumping boxes. Acknowledgment includes dividing content (if essential) and distinguishes it utilising OCR strategy.

1.1 Existing System

Existing frameworks incorporate content discovery from recordings, taking single casing at once. A few techniques thought about key edges. A few strategies utilized numerous edges alongside content following. Most existing strategies utilized content acknowledgment and identification utilizing fleeting spatial based techniques. Considering outline by casing devours much time. Furthermore, distinguishing proof of key edges needs insight for frameworks, and they look for the assistance of Decision rules. Existing frameworks additionally neglect to exploit using the connection between Tracking and Detection/Recognition and criticism between them is disregarded.

1.2 Proposed System

In this article, the Bayesian system of T^2DAR for inserted

inscription content which performs both following based content discovery and following based content acknowledgment in a solitary pipeline. The following by-identification is utilised to recover the missing location and input data between the following discovery and following acknowledgment is viewed as which has progressively exact outcomes.

2 Literature Review

Different researchers work accessible for separating content from the video captions. But the concepts for tracking the texts from the video are different for different researchers. Till now, there is no accurate method available to detect text from all types of videos. Here some of the works of literature studied to work in our proposed method. Shu Tian et al [1], given a new framework for text detection and recognition using the tracking the web videos. At first, the a unified Bayesian-based framework is applied for both tracking based text detection with recognition from complex web video. This framework is implemented within a Bayesian model exchanging the information between tracking and detection, which differs from the traditional strategies using knowledge-based rules. Also another approach is the tracking-by-detection for text tracking, in which the existing model used for region matching. The motion model for text tracking is adopted to connect the detections into trajectories. Another method is well formulated tracking based text detection and recognition approaches. Tracking based text detection uses feedback information between tracking and detection to retrieve the missing detections of the text and to report the issue of various text scales identifying the noises. Tracking based text recognition has over-segmentation and hierarchical clustering to select and combine recognized results in multiple frames. In their experiment, a real dataset is used for text detection and recognition from web videos, which includes complex web videos which are directly crawled from the publicly available in the web. Longfei Qin et al [2] implemented the categorization of video scene text frames for text detection and recognition. They developed a unified text detection and recognition method for different video types due to varying characteristics in the video. They proposed a new method to categorise various types of video text frames, namely, videos containing advertisement, signboard, license plate, the front page of book or magazine, street view, and video of general items, for better text detection and recognition rate. And they proposed symmetry features using gradient vector flow for Canny and Sobel edge images of each input frame to identify candidate edge components. Then for a candidate edge component image, we extract both global and local features using colours from different channels in a new way. The proposed method also extracts structural and statistical features from the spatial distribution of candidate pixels in a multi-scale environment. When they observe the images corresponding to the six classes, namely, advertisement, signboard, license plate, street view, book and other items, they we found that the colour feature can play a prominent role for classifying advertisement, book, license plate and items because generally the colour at text line level for these classes may not change much compared to its background. In the meantime, the colour of background may play a prominent role for classifying signboard and license plate because texts embedded in these classes of frames usually have a homogenous background. For a candidate edge component image, they further extract features based on colour histograms in grey, RGB and HSV colour spaces because it is known that these colour features play an important role in classifying the mentioned classes. They followed component selection feature extraction and next multiscale local and global feature categorisation. They have explored the common property of Canny and Sobel operation for identifying candidate edge components. For candidate edge components image, they extracted different features such as colour, statistical and spatial features at different scales: globally and locally. Xu-Cheng Yin et al. [3], presented a comprehensive survey of text detection, tracking and recognition in video with three major contributions. First, a generic framework is proposed for video text extraction that uniformly describes detection, tracking, and recognition with their relations and interactions. Second, inside this framework, an assortment of methods, systems and evaluation protocols of video text extraction are outlined, thought about, and broke down. Existing text tracking methods, tracking based detection and acknowledgment procedures are specifically featured. third, related applications, noticeable difficulties, and future directions for video text extraction (particularly from scene videos and web videos). In video content understanding, spatial-temporal analysis and multi-frame integration are general strategies. Compared to images, the temporal information (the dependencies between adjacent video frames) is helpful in improving text detection and recognition. Correspondingly, using spatial and temporal information acquired from multiple frames, video text tracking, tracking based detection, and tracking based recognition methods are comprehensively surveyed and highlighted in this section. They have used temporal spatial information-based methods for text tracking and detection. For recognition they have used image enhancement and fusion techniques. Ding Jie et al [4], implemented the edge analysis algorithm to extract overlaid text in the video. SVM is used to classify the pixels into text and non-text pixels from the HSV colour image. The V component of the image is used as a gray image. OCR extracts the maximum gradient feature of video text. The pseudo text area is removed after combining the connected domain by the morphology analysis. The extracted features of connected components and integration to improve the video text recognition. Their simulation results has shown a good detection effect in video text recognition. Jian Yi et al [5] has given a new approach for the multiple frame integration of video. It consists of three phases: in the text-block group (TBG) identification, the blocks with the same text are considered jointly for the location, edge distribution and contrast of the text block. In the TBG filtering, to maintain a strategic terrible impacts of the blurred text on the integration, they measured the clarity of the content using the text-intensity map for the clear text for the integration. The blurred text is filtered to avoid bad affects. The four direction text-intensity detectors are used to measure the text-intensity that correspond to the text strokes in the four directions i.e. horizontal, vertical, left diagonal and right diagonal directions respectively. The clear text has higher intensity than the blurred text in the text-intensity map, and the intensity of the text was implemented in the text-intensity map to gauge the clarity of the text in the image. They got the 57% precision and 60% recall and 8% repeat in their experimental methods in which they considered these as their best results by using TBG Filtering. Pooja and Renu Dhir [6], implemented video

OCR to extract overlay text and scene content. Usually, Text in video appears in the video for not more than a few seconds. Several text lines/characters remain unaffected for the duration of their lifetimes. In some videos,the text movement is simple and linear. It is a challenge to extraction of frames out of the videos itself. Text is to be extracted based on the segmentation of the word and the character extraction from background text line. Polina M. Osina et al [7], presented an algorithm for text recognition from images and videos. The algorithm was based on the combination of discrete cosine transform (DCT) and convolutional neural networks (CNN). In this paper, the description of the applying features of DCT for text detection is provided. Text-Attentional Convolutional Neural Network (Text-CNN) focuses on extracting text-related regions and features from the image components. In CNN detection step follows step word recognition. These two processes are not completely separated from each other, the information obtained in these steps are used to combine the recognition and the rank detection results. Their proposed algorithms can be implemented in text detection from real scene images and videos. The velocity of the video and the text has the important factor for the DCT coefficients to reperesent the text structure, which reflects a good text detection result. Moteelal and Murthy [8], used the location of the content for open-air and indoor wearable or handheld camera applications. In such situations, as the content of the sent to OCR or to a content to-discourse motor for computerised shape. In this paper, the authors presented a method of computation of keyframe using threshold of the positive difference of histogram. The threshold point is computed using the mean and standard deviation of positive difference of histogram to study the feature difference of consecutive video frames. Key-frame extraction is based on a complete difference of histogram of successive image frames. In this process, the threshold is calculated from the mean and standard deviation of the histogram of the total difference of consecutive image frames. Later the extracted key-frames are compare with the threshold contrary to an absolute difference of consecutive image frames.

3. Mathematical Functions Used in the Experiment

In this work, the experiment is conducted using MAC 64 bit Operating System with Intel i5 Processor, 4GB RAM, Open CV Software, Python 3.6 and Python-tesseract .

3.1 Bilateral Filter

The bilateral filtering function applies to the input image, that reduces unwanted noise keeping the fairly sharp edges. It is a non-linear, edge-preserving and noise-reducing filter. It replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels. However, it is very slow compared to most filters, but gives accurate results compared to other filters. The bilateral filter to an image is defined as

$$I^{filter}(x) = \frac{1}{W}\sum_{x_m \in \Omega} I(x_m) f_k\big(\|I(x_m)-I(x)\|\big) g_s\big(x_m-x\big) \quad (1)$$

where W is defined as,

$$W = \sum_{x_m \in \Omega} \|I(x_m)-I(x)\| g_s(x_m-x) \quad (2)$$

The terms in the equations are:

I^{filter} : the filtered image, I: the original input image, x: the coordinates of the current pixel to be filtered, Ω: the window centered in x, so $x_m \in \Omega$ is another pixel, f_k: the range kernel for smoothing differences in intensities for Gaussian function, g_s: the special kernel for smoothing differences of coordinates in Gaussian function, and W: the weight assigned using the spatial closeness for g_s and the intensity difference using the range kernel fk [12]. The Sigma values in the bilateral filter are, (i) if they are small (< 10), the filter will not have much effect, and (ii) if they are large (> 150), they will have a very strong effect, making the image look in different way. The large filters which are a diameter (d) of each pixel neighbourhood is used during filtering (d > 5) are very slow; practically, it is recommended to use d=5 for real-time applications, and d=9 for offline applications for heavy noise filtering.

3.2 Gaussian Blur

Gaussian Blur is used to blur an image by a Gaussian function. Generally it is used to reduce noise and the detail. Mathematically, Gaussian Blur is achieved by convolving the image with Gaussian Function [12], defined as:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{x^2}{2\sigma^2}} \quad (3)$$

where, x: the distance from the origin to the horizontal axis, y: the distance from the origin to the vertical axis, and σ: the standard deviation of the Gaussian distribution.

In two dimension, the above formula produces a surface of contours. These values are used to build a convolution matrix that will be applied to the original image [12].

Fig 1. *Impulse response of Gaussian*

For each pixel in the image the Gaussian function in two-dimension is blurs an image using a Gaussian filter. Applying the Gaussian Blur, the results are as follows (Fig.1):

Fig 2. *(a) Image before (b) Image after applying Gaussian Blur*

3.3 Converting the Image

In case of a transformation from RGB color space, the order of the channels should be specified either RGB or BGR. The R, G, and B channel normal ranges are:

(i) 0 to 255 for CV_8U images , (ii) 0 to 65535 for CV_16U images and (iii) 0 to 1 for CV_32F images.

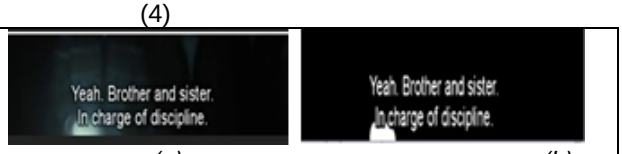
3.4 Threshold

The noise will be reduced by a fixed-level threshold to each array element of the image to extract the text easily. The fixed-level thresholding for a single-channel array with 8-bit or 32-bit floating point is used to get a bi-level/binary image out of a grayscale image or for removing the noise. Here, the

pixels are filtering out for too small or too large values. In this paper, a binary threshold is implemented [10](Fig 3). (here θ is the Threshold parameter)

Binary Threshold

$$d(x, y)=\begin{cases}MAX\_Value, & \text{if } src(x, y) > \theta \\ 0, & \text{otherwise}\end{cases} \quad (4)$$



(a) (b)

Fig 3 *(a) Before Thresholding (b) After Thresholding*

3.6 Sequence Matcher

Sequence Matcher supports a automatic heuristic treat on certain sequence items as junk. The heuristic counts is the number of times of each individual item appeared in the sequence. If an item's duplicates (after the first one) account for more than 1% of the sequence and the sequence is at least 200 items long and the item is marked as "popular" and is treated as junk for the purpose of sequence matching.

3.7 EAST Text Detection Tool

Easy and Accurate Scene Text detection pipeline (EAST tool) is a deep learning model based on new architecture and training pattern [15]. The key component of this model is a fully CNN that is adapted for text detection for the dense per-pixel predictions of words or text lines [1]. It eliminates, the transitional steps such as the candidate proposal, the text region formation and the word partitioning. The post-processing steps only include thresholding and NMS on predicted geometric shapes[10]. The following image shows the text detection output from EAST deep learning network (Fig 4).

Fig 4. *Image showing text detection in a frame from a sample video*

4. Components of the Proposed Method

Text detection from an image or an individual frame of videos is fundamental work to text tracking and recognition in video. Sliding window-based methods (region-based methods) is used to decide whether the image patch is text or not. Due to the multiple scales of scanning-windows, the computational power is high.

4.1 Text Tracking with Tracking-by-Detection (T3D)

Although Tracking Embedded caption text appears to be easy, as the location of text won't change across multiple frames, there exist challenges. The complex backgrounds and varied colours impose various challenges. Here, "tracking" isn't just essential yet additionally significant for detecting and recognizing embedded captions from WEB videos. In the text tracking, a basic advance is to persistently decide the location of text across multiple frames. Nonetheless, this methodology isn't a trifling task however embedded caption text is apparently still in precisely the same position in each frame. Initially, in light of difficulties with complex backgrounds and varied colors, text detection results (region locations) for one same caption text are not totally fixed (still), in any case, consistently with several pixels translation. All the more critically, caption text switching progressively shows up, consistently with a similar background and similar text. It is truly testing to recognize the change (cutting) of such captions in nonstop frames. In addition, on account of difficulties with complex backgrounds, shifted colors, comparative colors and low contrast, it is likewise not a simple assignment to identify two text regions in consecutive frames whether they are same or not by straightforward locale matching strategies. Thus, "tracking" has been as of now presented for detecting and recognizing caption text. Similarity calculation is performed for the detection and decomposed into appearance similarity and location similarity.

4.2 Tracking Based Text Detection (TBTD)

A part of the detection of text, there are noises on text regions. These texts may not match with the trajectories partly because of the variations of backgrounds which are always complex which shows the similarity between the detection and the last frame of the trajectory is low. In this experiment, the apparent similarities between the detected result considered as noise and the text regions of all trajectories in the last 10 frames are calculated. A new trajectory is initialized if three consecutive frames are similar and is terminated if it does not detect any similarity across five consecutive frames. Some detection some times wrongly classifies as noise during the initialization. Using tracking these detected texts can be correctly classified by modified detection method. The presence of similarities are considered around neighbouring 10 frames and if it exceeds a threshold (usually 0.7), then the text region is considered to retrieve the missing detection.

4.3 Tracking Based Text Recognition (TBTR)

The tracking trajectories are incidentally over segmented to check if detections relate to a similar text. The tracking trajectories are transiently over-segmented to guarantee that the detections of each sub-trajectory are relating to a similar text. An agglomerative hierarchical clustering algorithm is utilized to merge over-segmented trajectories and build suitable trajectories, i.e., these developed trajectories have a moderate length for multi-frame coordination. Finally, a voting technique is directed to acquire the last recognition results with these reasonable trajectories. The tesseract-OCR is used to recognize the text and the words in the individual frames. The results with high confidence (>0.65) are considred reated as the actual text.

4.4 Over Segmentation and Merging

Every trajectory is freely and transiently over-segmented. In this fleeting over-segmentation step, the perceived text with high confidence is viewed as the reference text, and the frame with the reference text is viewed as a reference frame. The acknowledgment brings about different frames are viewed as the noise text. The trajectory is segmented into a few short sub-trajectories dependent on the reference frame. The limit of two segmentations is resolved to be the median frame between two progressive reference frames. Correspondingly, the detections on the frames between two progressive boundaries make each sub-trajectory. Clearly, all content from one trajectory is considered as a similar one in text tracking. Also, the parameters of temporal segmentation are tuned to

guarantee all original content of each sub-trajectory is nearly a similar one. Thus, the division of one trajectory is probably going to be over-divided, which isn't immaculate however superior to under-division where the ID switch consistently happens. To additionally improve the exhibition of image segmentation, a cluster algorithm is then used to combine the sub-trajectories with a short length. This merging procedure is performed by an agglomerative hierarchical clustering algorithm. The trajectory's highlights are gotten from the reference text. The dissimilarity between two highlights is estimated with the edit distance. During the time spent clustering, the separation between two (break) clusters is determined on two agent focuses (one in each cluster) which are the nearest pair focuses in the two (interval) clusters. The temporal information isn't utilized in the consolidating step in light of the fact that a similar content may not happen constantly since the mistake of the ID switch is probably going to show up in text tracking. In the wake of combining, the trajectories of an unassuming length with high certainty are created and refreshed.

4.5 Image Processing

Some image processing operations on the image are performed to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is an image [9]. The purpose of image processing is divided into 5 sets. They are:

1. Visualization - Observe the objects that are not visible.
2. Image sharpening and restoration - To create a better image.
3. Image retrieval - Seek for the image of interest.
4. Measurement of pattern – Measures various objects in an image.
5. Image Recognition – Distinguish the objects in an image.

5. CONSIDERATIONS

The following concepts are considered for text detection from the videos.

5.1 Frame Rate

Frame rate is the number of individual video frames that the camera captures per second. Frame rate comes in a few different standards (expressed as frames per second or fps): 24fps, 25fps, 30fps, 60fps, and 120fps.

5.2 Human Vision

The temporal sensitivity and resolution of human vision shifts relying upon the sort and characteristics of visual stimulus, which varies between the individuals. The human visual system can process 10 to 12 images for each second as individuals, while higher rates are seen as motion. As to image recognition, a person can recognise a specific image in a robust series of various images, every one of which endures as meagre as 13 milliseconds.

5.3 Work Flow

This work focuses on extraction Caption Text from videos. This requires the following tasks to be done. Given a video as input, frames are extracted, text regions are detected, and text is extracted. Those steps involved are shown in the following workflow:

Step-1: Take a video running at 13fps to 24fps framerate. This is the standard framerate range where most of the videos fall into. The following image is a snapshot taken while the video is playing(Fig 5).

***Fig 5**. A frame of a video*

Step-2: Extract the frames from that video. VideoCapture() is the method available in OpenCV to extract frames. The extracted frames will be(Fig 6):

(a) frame390.jpg *frame400.jpg*

***Fig 6**. Frames after extraction*

Step-3: Feed the extracted frames to EAST Text Detection Tool. The result will be an image with a bounding box surrounding the textual content present in the frame. The detection results of EAST Tool fail when the background is complex or the colour of background and caption is similar. After detection, the image will be as follows(Fig 7).

***Fig 7**. Result of Text Detection showing bounding box*

Step-4: These images are cropped along the bounding box. The box coordinates are acquired from the EAST Text Detection output (Fig 8).

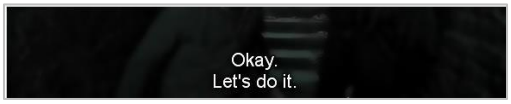


***Fig 8**. The Cropped Image*

Step-5: These cropped images are preprocessed using the pre-processing methods provided in OpenCV libraries. The purpose of pre-processing is to enhance the image so that the details are highlighted much and the interest of the image can be better focused on. The general pre-processing techniques used are Filtering, Blurring, Thresholding (Segmenting the image) etc. After pre-processing, the image will be as follows (Fig 9):

Fig 9.*Cropped Image on pre-processing*

Step-6: The pre-processed frames are fed to PyTesseract which uses OCR for recognising the text inside the image. OCR recognises the text character by character. The extracted text is give below (Fig 10):

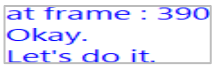


Fig 10. *OCR Output-Extracted Tex*

6. Results

From the Given a video as input, the frames are extracted as follows (Fig 11):

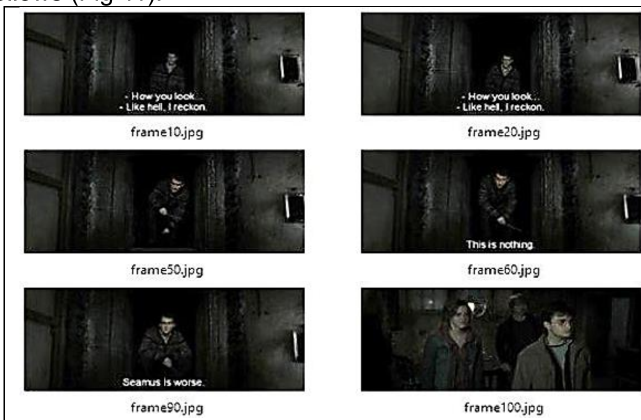


Fig 11*. Extracted Frames from the video*

And then, after the text detection, the cropped regions containing text are acquired as follows(Fig 12):

Fig 12*. Cropped Frames*

The following are the results of text extraction from 2 sample videos (Fig 13):

(a) frame30.jpg

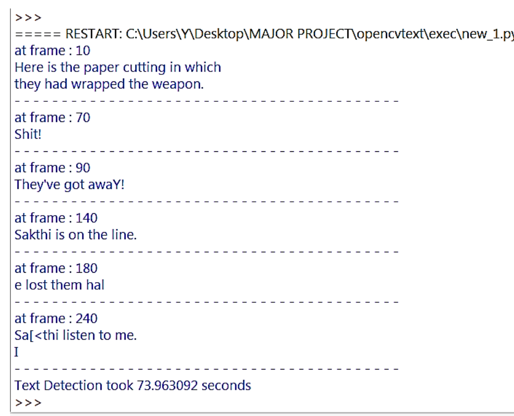
(b) frame110.jpg

(c) frame130.jpg

(d) frame200.jp

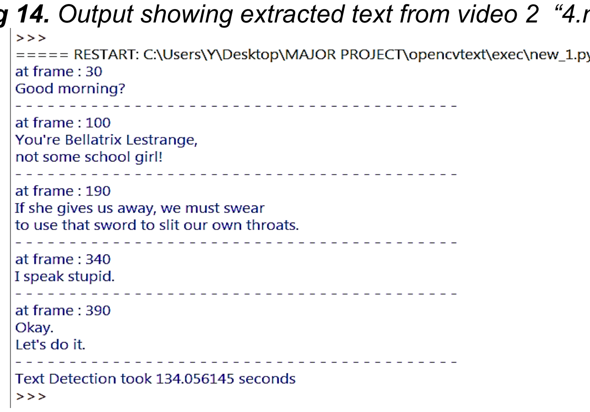
Fig 13. *Text Extraction from Frames*

The following are the results of different text extraction from other two sample videos(Fig 14 and Fig 15):



```
>>>
===== RESTART: C:\Users\Y\Desktop\MAJOR PROJECT\opencvtext\exec\new_1.py
at frame : 10
Here is the paper cutting in which
they had wrapped the weapon.
-----------------------------------------
at frame : 70
Shit!
-----------------------------------------
at frame : 90
They've got awaY!
-----------------------------------------
at frame : 140
Sakthi is on the line.
-----------------------------------------
at frame : 180
e lost them hal
-----------------------------------------
at frame : 240
Sa[<thi listen to me.
I
-----------------------------------------
Text Detection took 73.963092 seconds
>>>
```

Fig 14. *Output showing extracted text from video 2 "4.mp4"*



```
>>>
===== RESTART: C:\Users\Y\Desktop\MAJOR PROJECT\opencvtext\exec\new_1.py
at frame : 30
Good morning?
-----------------------------------------
at frame : 100
You're Bellatrix Lestrange,
not some school girl!
-----------------------------------------
at frame : 190
If she gives us away, we must swear
to use that sword to slit our own throats.
-----------------------------------------
at frame : 340
I speak stupid.
-----------------------------------------
at frame : 390
Okay.
Let's do it.
-----------------------------------------
Text Detection took 134.056145 seconds
>>>
```

Fig 15. *Output showing extracted text from video 3 "hp1.mp4"*

7. Analysis of Results

For video-1, at frames 200, 240, 380, 420 and 480; and for video-2, at frames 140 and 180 there is a major deviation in the graph, due to difference in percentage match with cropped and actual frames. This happened because at those frames, Text Detection failed due to background complexity. In some cases, the Text Detection (EAST) network (pre- trained) failed to detect the text. When the network is analyzed, and the detection capability is enhanced by modifying the network, the

detection results will be much better, resulting in improved accuracy.

The details of the tables have shown the respective accuracies for cropped and actual frames:

1) Video 1: (Table 1)
2) Video 2: (Table 2)
3) Video 3: (Table 3)

From the tables the graphical representations are given below.

1) Video 1: (Fig 16)
2) Video 2: (Fig 17)
3) Video 3: (Fig 18)

For both the tables and the figures, refer the Appendix.

8. Conclusion and Future Scope

The both detecting and Recognizing text in the video shares a few difficulties practically speaking, for example, robustness to background complexity, text degradation and distortion, content variations, and moving objects. The text background is regularly intricate, particularly for scene message and embedded caption text. A few parts of the background can be fundamentally the same as the text and text objects are normally little, the two of which lessen the accuracy of text detection and tracking. Pre-preparing the image aides, and text recognition is effective with improved accuracy. Likewise, the east text detector should be broke down and altered further to improve its text detection yield, for it to be utilized in progressively practical and complex applications. As the extraction text is stored in this work, this can be further extended in the automation system. The program and algorithm be fed to automated devices which can be giving intuitive services to illiterates, disabled as well as they process the data faster and can convert it as speech, presentation etc. and serve as digital video analyzer assistants.

References

[1] Shu Tian, Xu-Cheng Yin, Ya Su, Hong-Wei Hao, "A Unified Framework for Tracking based Text Detection and Recognition from Web Videos", IEEE Transactions on Pattern Analysis and Machine Engineering, Vol. 40, No. 3, March 2018.

[2] Longfei Qin, Palaiahnakote Shivakumara, Tong Lu, Umapada Pal and Chew Lim Tan, "Video Scene Text Frames Categorization for Text Detection and Recognition", IEEE International Conference on Pattern Recognition, December 4-8, 2016.

[3] Xu-Cheng Yin, Ze-Yu Zuo, Shu Tian and Cheng-Lin Liu, "Text Detection, Tracking and Recognition in Video: A Comprehensive Survey", IEEE Transactions on Image Processing, vol. 25, no. 6, pp. 2752-2773, June 2016.

[4] Ding Jie, Zhao Guotao, Xu Fang, "Research on Video Text Recognition Technology Based on OCR", IEEE 10th International Conference on Measuring Technology and Mechatronics Automation(ICMTMA), Changsha, pp. 457-462, 2018.

[5] Jian Yi, Yuxin Peng, Jianguo Xiao, "Using Multiple Frame Integration for the Text Recognition of Video", IEEE 10th International Conference on Document Analysis and Recognition, Volume: 1, Pages: 71-75 2009.

[6] Pooja, Renu Dhir, "Video Text Extraction and Recognition: A Survey", International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, pp. 1366-1373, 23-25 March 2016

[7] Polina M. Osina, Yuliya A. Bolotava, Vladimir G. Spitsyn, "Text Detection Algorithm on real scene images and videos on the base of Discrete Cosine Transform and Convolutional Neural Network", International Siberian Conference on Control and Communications (SIBCON),2017.

[8] T.Moteelal, V.Sreerama Murthy, "Text Detection from a Video Using Frame Extraction and Text Tracking", IEEE International Conference on Intelligent Sustainable Systems (ICISS 2017), pp. 457-461., 7-8 Dec. 2017.

[9] Rafael C. Gonazalez, Richard E.Woods, "Digital Image Processing", 3rd Edition, Pearson, 2001.

[10] Xinyu Zhou et al, "EAST: An Efficient and Accurate Scene Text Detector", arXiv:1704.03155v2 [cs.CV] 10 Jul 2017

[11] https://docs.opencv.org/3.4/index.html, 2019

[12] https://libraries.io/pypi/bootstrap-difflib, 2019

[13] https://docs.python.org, 2019

[14] http://www.dai.ed.ac.uk/CVonline/LOCAL_COPIES/MANDUCHI1/Bilateral_Filtering.html,

APPENDIX

MATCHING TABLES FOR EXTRACTED TEXTS FROM VIDEO FRAMES

TABLE 1

VIDEO-1"HP.MP4"

Frame Number	% match with Cropped frame	% match with Actual frame
Frame 30	100	100
Frame 110	100	100
Frame 130	100	100
Frame 200	95.6	69.6
Frame 240	96.2	81.8
Frame 280	91.3	91.3
Frame 340	87.5	80.7
Frame 380	95.8	47.9
Frame 420	100	16.6
Frame 460	100	100
Frame 480	100	78.5
Frame 530	92.5	92.5
	Avg: 96.5	Avg: 80.35

TABLE 2

VIDEO-2 "4.MP4"

Frame Number	% match with Cropped frame	% match with Actual frame
Frame 10	100	100
Frame 70	100	100
Frame 90	92.3	92.3
Frame 140	100	80.95
Frame 180	100	50
Frame 240	93.75	93.75
	Avg: 97.6	Avg: 86.16

TABLE 3

VIDEO- 3 "HP1.MP4"

Frame Number	% match with Cropped frame	% match with Actual frame
Frame 30	100	100
Frame 100	100	100
Frame 190	100	100
Frame 340	100	100
Frame 390	100	100
	Avg: 100	Avg: 100

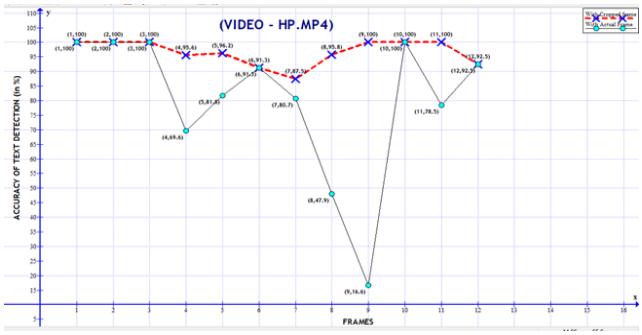


Fig 17. *Video-1 "hp.mp4"*

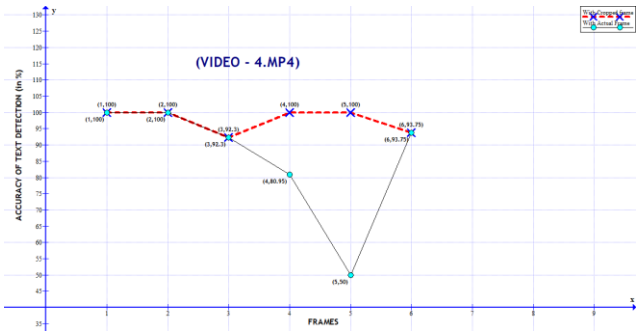


Fig 18. *Video-2 "4.mp4"*

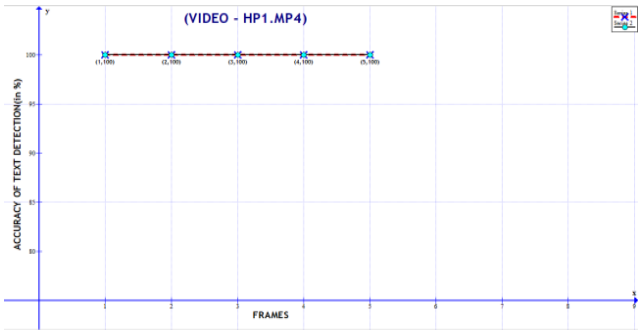


Fig 19*. Video- 3 "hp1.mp4"*