IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 24, Issue 6, Ser. I (Nov. – Dec. 2022), PP 49-60
www.iosrjournals.org

A Novel Hybrid Architecture For Accurate Text Detection In Video With Text Skeletonization And Reinforced Frcnn

Mortha Manasa Devi[1] , Dr.Maddala Seetha[2], and Dr.S.Vishwanadha Raju[3]
1 Ascent FS India (LLP), India.
2 Department of Computer Science &amp; Engineering,
G.Narayanamma Institute of Technology and Science, Hyderabad, 500008, India
3 Department of Computer Science &amp; Engineering,
Jawaharlal Nehru Technological University, Jagityal, 500085, India
** Corresponding Author: Mortha Manasa Devi. Email: morthamanasa@gmail.com*

Abstract: *Video text extraction plays an important role in multimedia understanding and retrieval. Most of the previous research efforts are conducted within individual frames. In this paper, a novel solution based on hybrid architecture is proposed, which employs both the FRCNN and Skeletonization to provide a consistent end-to-end trainable text identification system. Object detection is made faster and more precise with the Faster R-CNN, which is an end-to-end CNN. While the original FRCNN utilizes the feature map from the final convolutional layer to generate regions of interest (RoI), the suggested method incorporates the skeletonization process to detect the specific text in the video frames. Results obtained from experiments performed on picture frames from a video dataset show that the given hybrid method based on FRCNN and skeletonization has increased detection scores while maintaining nearly the same detection speed when compared to other recent approaches.*

Key Words: *Text Detection, Skeletonization, Deep Learning, Data Augmentation, Localization, and Labelling.*

Date of Submission: 16-12-2022 Date of Acceptance: 31-12-2022

I. INTRODUCTION

Nowadays detection of textual information from videos is a very challenging and exciting research area in the video processing and machine learning field (Y. Liu et al., 2019). In addition to the audio-visual content, the text present in the video images provides an important clue that can be exploited for developing video indexing and retrieval systems (Jamil et al., 2019). The escalating popularity of digital camera devices and smartphones has led to the explosive growth of scene video data in daily life (X. Wang et al., 2019). There are a variety of categories of video material (TV news, documentaries, movies, CCTV, etc) (Manita et al., 2021). Many techniques have been developed to detect and recognize the text in scene images and video. (X. Liu et al., 2019). Traditional text detection methods mainly use extremal region, border information, or character’s morphological information to locate text (Zhu & Du, 2021), and then employ variant tracking algorithms to propagate text candidates across frames and match them with detected ones for improved detection accuracy and completeness (L. Wang et al., 2019). Compared to text appearing in natural scene images, scene text in the videos has some common attributes such as varied appearances and complex context, which bring extra difficulties to the reliable detection of text in the videos (Y. Wang et al., 2019).

1.1. Problem definition

Although text detection finds a vital role in current applications, the text region detection from a video has several challenges as follows,

- Due to the diversity of scene text’s appearance and quality, reliably localizing scene text in complex video contexts remains a highly challenging task.
- However, the existing approaches took a prolonged time to execute, and sub-optimal solutions and the efficiency of methods is not satisfactory.
- The text edges of text parts are less clear than in natural scene texts. This causes misclassification of the foreground in low-contrast images and inaccurate localization.

These are the key points motivating them to propose a hybrid architecture, to provide an end-to-end consistently trainable text identification system. Therefore, the main contributions of the proposed model are,

- To improve the quality of the frames by employing hybrid pre-processing techniques.
- To progress the model performance through feature extraction and data augmentation with localizing and labeling the regions of text for generating precise ground truths.
- To utilize the RPN-FRCNN prediction model and optimize it by integrating the proposed approach and analyzing the accuracy.
- To develop a framework to deploy the proposed model for displaying converted video with bounded text.

The following sections of the paper are written as follows: Section 2 reviews the previously developed models related to text detection. Section 3 explains the proposed text detection approach. Section 4 presents the results and discussions. The last section concludes the paper.

II. RELATED WORK

(Huang, 2019) developed a new approach to detect video scene text based on a saliency edge map. The complex background removal and detection of text with low resolution were done by retrieving and integrating the saliency map and edge map. Finally, the Gaussian mixture model (GMM) was used to get the text regions. Experimental evaluations demonstrated that the method outperformed the other text detection algorithms. Post-processing was often inevitable to improve performance.

(Guan et al., 2022) implemented a refined feature attentive network (RFN) to solve the inaccurate localization problem. First, the feature integration mechanism constructed an adaptive feature representation and an attentive proposal refinement module rectified the location deviation of candidate boxes, in addition to a re-scoring mechanism to select text boxes. The method achieved state-of-the-art performance. The accuracy rates of the model need to be further improved.

(Y. Liu et al., 2020) improved text detection performance using Mask Tightness Text Detector (Mask TTD). Mask TTD used a tightness prior and text frontier learning to mask prediction. In addition, it integrated a branch for the polygonal boundary of each text region, to improve the detection performance. The Mask TTD achieved state-of-the-art performance. The method brought large false positives, which suggests using efficient post-processing techniques to improve the detection performance.

(Xue et al., 2019) detected texts in blurred/nonblurred images. The method first estimated the degree of blur based on neighbor gradient values. Then performed K-means clustering for separating text pixels from non-text ones and symmetry features were extracted using Bhattacharyya distance. Then, the method found the nearest neighbor for each text component to fix the bounding box for the whole line. Experimental results showed that the method outperformed the existing methods. The method's result required the design of a proper mechanism to optimize the blurred and non-blurred issues present in the images.

(Cai et al., 2020) detected video scene texts in both spatial and temporal domains with a new large-scale benchmark, named STVText4, spatial-temporal detection metric (STDM), and a novel clustering-based baseline method, referred to as Temporal Clustering (TC). Text instances in STVText4 were annotated with both spatial bounding boxes and temporal ranges. Experiments demonstrated the efficacy of the method. The system lacked a mechanism to select an appropriate parameter set.

(Feng et al., 2021) suggested an end-to-end trainable video text detector that tracks texts based on semantic features. Semantic features were extracted with a new character center segmentation branch. Then, an appearance-semantic geometry descriptor and a weakly-supervised character center detection module were introduced to track text instances and generate character-level labels. The method achieved state-of-the-art performance. However, character center segmentation was complex for separating the text instances close to each other.

(Zhang et al., 2020) explored a novel unified relational reasoning graph network for arbitrary shape text detection. The text instance was divided into rectangular components, and the geometry attributes were estimated by the text proposal model to establish linkages. The local graph construction model was used for further reasoning and deducing the likelihood of linkages. Experiments demonstrated the state-of-the-art performance of the method. The method easily brought numerous non-text components and filtering out the false positives were critical to the success of the method.

(Cheng et al., 2021) implemented a fast and robust end-to-end video text spotting framework (FREE). FREE united the detector and recommender into a whole framework and helped to achieve global optimization. A detector learned text locations among video frames and selected the highest-quality text from text streams using a novel text recommender. Extensive experiments showed the method achieved remarkable state-of-the-art. Referring to the temporal relationship of more texts in consecutive frames was a time-consuming process.

(Bhunia et al., 2019) presented a CNN-LSTM framework for script identification. The model computed the attention-based patch weights for the image patches using the softmax layer after LSTM and the local features and Global features were yielded and fused with patch-wise multiplication of those weights and from the last cell state of LSTM. Then a fusion technique was employed to weigh both features. Training of detection framework with discriminative features relied upon increased computation complexity.

(Yu et al., 2021) integrated video text detection and tracking in a unified framework through the descriptor generation module. The anchor-free regression was adopted to detect the quadrangles of words in a per-pixel manner. The detected proposals and common feature maps of the current frame were fed into the descriptor generation module for tracking. The method significantly outperformed state-of-the-art methods. The trajectory generation incorporated in the method was not at the best possible level of performance.

III. PROPOSED TEXT DETECTION FRAMEWORK

Initially, the input videos are converted into frames and the quality of each frame is improved by using hybrid preprocessing techniques. Hence, the preprocessed frames are skeletonized by using the enhanced thinning algorithm to extract the general form of the text. Then, the adaptive Text Feature Extraction and Data Augmentation with enhanced localization process are carried out. Finally, detection of text through FRCNN architecture by adapting dynamic bounding box generation is done. The general flow of the proposed model is depicted in the following figure 1,

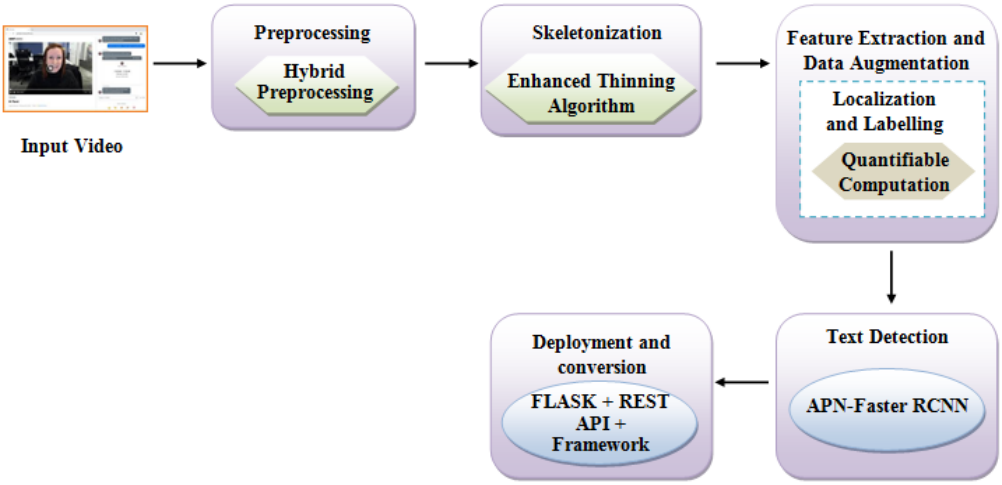
Input Video
Preprocessing
Hybrid Preprocessing
Skeletonization
Enhanced Thinning Algorithm
Feature Extraction and Data Augmentation
Localization and Labelling
Quantifiable Computation
Text Detection
APN-Faster RCNN
Deployment and conversion
FLASK + REST API + Framework

Figure 1: Block diagram of the proposed model

3.1 Preprocessing

Initially, the input video taken for the proposed model is converted into a number of frames and is applied for preprocessing to improve the quality of the frames. In this work, preprocessing techniques are employed to obtain a better text detection rate. The number of frames is signified as,

$$Input\ video : V_i \rightarrow fr_{(m)} \tag{1}$$

$$fr_{(n)} = \{ fr_{(1)}, fr_{(2)}, fr_{(3)}, ...I_{(N)} \} \tag{2}$$

Wherein, $fr_{(n)}$ denotes the sequence frameset and $fr_{(N)}$ implies the N -th number of frames, and V_i is the input video to obtain the number of frames. The frames are further applied for the following preprocessing steps,

Greyscale conversion: This is the step to convert the input frames into grayscale so that the hue and saturation information in the images are removed. It can be expressed as,

$$fr_{gr(n)} = 0.298R + 0.587G + 0.114B \tag{3}$$

Where, $fr_{gr(n)}$ is the converted frames and the coefficients of red, green, and blue colors are denoted as, R, G, B.

Noise Removal: Morphological operations were used to delete small noises and enhance images by ensuring that no part of the text is removed.

Segmentation: Segmentation is performed based on pixel level by taking threshold value to remove background and get all occurrences of text present in the image into meaningful units and make the segments of those units such as characters, words, and text lines.

Smoothening enhancement: The rotational gradients are used for enhancing low-contrast structures on the images. The gradient removes the object with a rotational symmetry with respect to pixel coordinates. Then, a combination of erosion and dilation operations is used to remove small empty holes and clear outliers. Finally, the smoothened frames are obtained as,

$$fr_{sm(n)} = fr_{(n)} \circ (1 - \alpha * \nabla(fr_{(n)}) \tag{4}$$

Where, $fr_{sm(n)}$ denotes the resultant frame, $\circ$ is the element-wise multiplication operator, α is the smoothing factor to determine how much information needs to be removed on those regions that are considered perturbed, $\nabla(\bullet)$ is the gradient magnitude map.

Similarly, text filling and intensity behavior analysis are implemented to improve the quality of frames. Hence, the preprocessed frames $\left(fr_{pre(n)}\right)$ are expressed as,

$$fr_{pre(n)} = \{ fr_{pre(1)}, fr_{pre(2)}, fr_{pre(3)}, \ldots\ldots\ldots\ldots fr_{pre(N)} \} \tag{5}$$

3.2 Skeletonization

In this step, the skeleton of all preprocessed images is extracted. Skeletonization is a kind of thinning process where the image is scanned to determine the pixel to be removed or not. By doing so, a region-based shape feature representing the general form of the text was extracted. In this work, the enhanced Zhang-Suen thinning algorithm is used for this purpose. The algorithm consists of two sub-iterations processes aimed at removing all the contour points of the character except those points that belong to the skeleton. Since the above approach cannot guarantee one-pixel thickness for the thinned character, the proposed algorithm considered pixels in all angles to make sure it gets a complete text structure.

In the preprocessed image $fr_{pre(n)}$, for the value of the pixel $fr_{pre(n)}(x)$ with $x = (x_i, x_j)$, the neighborhood of the pixel in a different angle can be defined as,

$$G(x) = \left\{ y \in D \,|\, \sin\frac{\pi}{180^\circ} \lfloor \max |x_i - y_i|, |x_j - y_j| \rfloor \le 1 \right\} \tag{6}$$

Where, the pixels x, y are neighbors if they share an edge or vertex i.e., $y \in G(x) \Rightarrow x \in G(y)$, G denotes the neighborhood of the pixel in an image of a two-dimensional rectangular lattice D. Each pixel except the ones located at the margin and corner of the image has eight neighboring pixels.

For the first iteration, the algorithm search for the pixel that satisfies the following features such as,

$$2 \leq fr_{pre(n)}(x) \leq 6 \tag{7}$$

$$B(x) = 1 \tag{8}$$

$$x_{i-1,j} \times x_{i,j+1} \times x_{i+1,j} = 0 \tag{9}$$

$$x_{i,j+1} \times x_{i+1,j} \times x_{i,j-1} = 0 \tag{10}$$

Where, $fr_{pre(n)}(x)$ denotes the sum of non-zero (foreground) pixels in $G(x)$, $B(x)$ denotes the number of times the value of pixels $fr_{pre(n)}(x)$ changes from 0 to 1. The above-mentioned feature is used to remove the pixels at the extreme point, avoid cutting the main elements of the skeletonized image, and remove the bottom, right boundary, and upper-left corner pixels of the image. Hence, the pixels with such features are changed into background pixels.

Similarly in the second sub-iteration, the pixels that need to be removed are identified with the following conditions are,

$$2 \leq fr_{pre(n)}(x) \leq 6 \tag{11}$$

$$B(x) = 1 \tag{12}$$

$$x_{i-1,j} \times x_{i,j+1} \times x_{i,j-1} = 0 \tag{13}$$

$$x_{i-1,j} \times x_{i+1,j} \times x_{i,j-1} = 0 \tag{14}$$

During this stage, the pixels at the leftmost and top extreme point are removed, by keeping the main elements of the skeletonized image that satisfies none of the conditions. In this way, all the contour points of the character except those points that belong to the skeleton are removed to refine the text shape that needs handling for the purpose of training purposes.

3.3 Feature Extraction and Data Augmentation

This phase is for analyzing the Bounding boxes for the skeletonized images $fr_{ske(n)}$ where a large number of pixels of the image are efficiently represented in such a way that interesting parts of the image are captured effectively. In this work, feature extraction is executed with enhanced localization and labeling techniques.

Localization and Labelling: The objective of text localization is to place rectangles of varying sizes covering the text regions. To achieve this, the presented model employed analysis of geometric maps using quantifiable computation to identify the text components and group them to localize text regions. The presented model makes calculations on the pixel level to extract and define rectangle bounding boxes around the text regions.

Let the left top and right bottom points of the target bounding box in output coordinate space be defined as, $p = (x_i, y_i)$ and $q = (x_j, y_j)$. Hence, each pixel located at (x_m, y_m) in the output feature map describes a bounding box as,

$$a_m = \{g, r_{x,a} = x_m - x_a, r_{y,a} = y_m - y_t, r_{x,j} = x_m - x_j, r_{y,j} = y_m - y_j\} \tag{15}$$

Where, g is the confidence score of being a text, $r_{x,a}, r_{y,a}, r_{x,j}, r_{y,j}$ are the distance amid output pixel location and the boundary of a target bounding box.

Finally, each pixel in the output map is converted to the bounding boxes to localize the text objects. This semantic geometry has the benefit of training data labeled to include semantic orientation and provides an additional helpful signal to the detection network.

Then, to improve the performance of the detection model, data augmentation has been incorporated to find inputs in the form of extra feature information. In this work, the augmentation techniques such as shear angles, horizontal, and vertical angles are considered to increase the dataset and to make sure text is detected in an angle. Thus, the output maps with augmented information are subjected to the text detection phase.

3.4 Text Detection

This phase is where the text detection is done by using the proposed Adaptive Proposal Network based Faster RCNN (APN-Faster RCNN). Faster R-CNN, an extension of fast R-CNN is one of the most well-known object detection neural networks. It is the combination of three sub-networks: feature extraction networks, Region Proposal Networks, and fast R-CNN. However, the conventional Faster R-CNN method has not achieved promising enough results due to two specific difficulties of text detection, i.e., texts in images have various sizes and scales, and this receptive field is sometimes too large or too small to detect texts. In this case, the selective search using the RPN cannot generate ROIs because of the insufficient feature. To overcome such an issue, the dynamic bounding box generation technique is adapted to avoid standard features existing from RPN-FRCNN. The architecture of the proposed APN-Faster RCNN is shown in below figure 2,

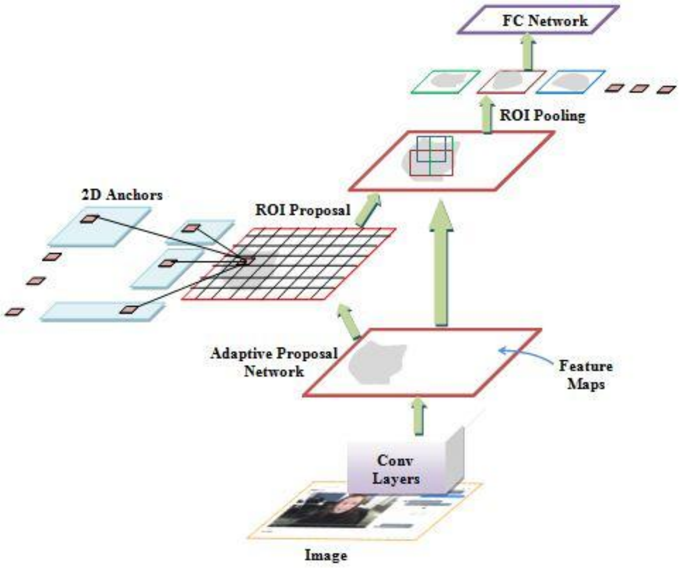

Figure 2: Architecture of APN-Faster RCNN

Feature Extraction Network: In this module, CNN has been used as a feature extractor. The function of this network is to generate good features from the images. Thus the feature map, which maintains the shape and structure of the original image, from the last convolution layer is forwarded to the RPN and classification network.

APN module: The purpose of APN is to generate region proposals. In this work, the dynamic bounding boxes are generated by setting the initial coordinates of the skeleton image. By using coordinates of skeleton key points as setpoints, the dynamic bounding box can move according to the camera movement, and the size of the bounding box can scale up automatically according to the text size and its position from the camera. The module gives the probability that a bounding box is an object or not.

Detection Module: The Detection Network takes input as the feature map from the first part and a set of region proposals from the second part. The feature map is cropped at a specific point and the ROI pooling layer utilizes the object proposals to extract a fixed-length feature map.

Then, the feature vector (f_i) is fed into the fully connected layer where the softmax probability is used to generate the object classes and bounding box positions for one of the classes. The softmax function $\psi_{sf}(f_i)$ is computed as,

$$\psi_{sf}(f_i) = \frac{\exp(f_i)}{\sum \exp(f_i)} \tag{16}$$

The fully connected layer performs parallel tasks such as regression and classification for refining the prediction boxes from the proposal boxes and obtaining the text detection probabilities. The output obtained from the fully connected layer is denoted as,

$$\Theta_{fcl} = \{\partial_{cls}, \partial_{rbb},\} \tag{17}$$

Where, Θ_{fcl} denotes the output of the fully connected layer, which contains bounding box coordinates for each object ∂_{rbb}, and the class of the object ∂_{cls} .

From the output of the network, the loss function is evaluated as,

$$l = l_c(\wp, \wp') + \varpi\wp' l_r(R, R') \tag{18}$$

Where, R is the probability of anchor box containing objects, R' is the ground truth labels, $\wp$ and $\wp'$ are the regression vector of the anchor box and its corresponding ground truth, ϖ denotes the weight balancing parameter, l_c and l_r are the loss functions of classification and regression. In the cases of the maximum loss function, the network needs to be trained for updating the weight parameters. In this work, the SGD optimizer is the employer for training the network.

Training: During training, the weights are optimized to minimize the loss value using an SGD optimizer, which in turn makes the Faster R-CNN detect the position of text regions more accurately. The parameter update using SGD for each training example $\{e_i\}$ can be expressed as,

$$\phi_{opt} = \phi - \beta\left(\frac{d(F(\phi; e_i, y_i))}{d\phi}\right) \tag{19}$$

Where, ϕ is the model parameter, y_i is the label, $F(\bullet)$ denotes the objective function, β denotes the learning rate. Algorithm 1 shows the pseudocode of the entire framework,

Algorithm 1: Pseudo code of the proposed model

```
                         Input: Video $V_i$
                     Output: Detected Texts

        Initialize number of videos $V_i$
                    $Input\ video : V_i \rightarrow fr_{(m)}$        // frames conversion
        For each frame do
            Apply hybrid preprocessing
            For pixels belongs to $fr_{pre(n)}$ do        //Skeletonization
                For odd iterations do
                    If $(2 \leq fr_{pre(n)}(x) \leq 6)$
                    {
                        Remove pixels at extreme points
                    Else if ( $B(x) = 1$ ){
                        Keep main elements
                    Else if $(x_{i-1,j} \times x_{i,j+1} \times x_{i,j-1} = 0)$
                        Remove bottom and right boundary
                    Else $(x_{i,j+1} \times x_{i+1,j} \times x_{i,j-1})$
                        Remove upper-left corner pixels
                    }
                    End if
                End for
                For even iterations do
                    If $\begin{pmatrix} 2 \leq fr_{pre(n)}(x) \leq 6 \\ B(x) = 1 \\ x_{i-1,j} \times x_{i,j+1} \times x_{i,j-1} = 0 \\ x_{i-1,j} \times x_{i+1,j} \times x_{i,j-1} = 0 \end{pmatrix}$ then
                    {
                        Remove leftmost and top extreme pixels
                    }
                    End if
                End for
            End for
            For extracted skeletons do        //feature extraction & data augmentation
                {
                Define bounding boxes around the text regions
                Mark bounding boxes with the class labels
            }
            End for
            Perform data augmentation
            Detect text regions using APN-Faster RCNN
        End for
        Return output videos with detected texts
    End
```

3.5 Deployment and Interaction

The whole system is developed with user interaction and ease of use in mind. To make the systems more accessible, the system was built as a web app using the FLASK framework so that it can be easily deployed.

IV. RESULT AND DISCUSSION

In this section, the performance of the proposed text detection system is analyzed. The proposed work is implemented in the working platform of PYTHON.

4.1 Performance Analysis

In this section, the performance of the proposed APN-Faster RCNN network is analyzed with the existing R-CNN, Fast RCNN, and faster RCNN methods based on IoU, prediction time, detection accuracy, and loss function.

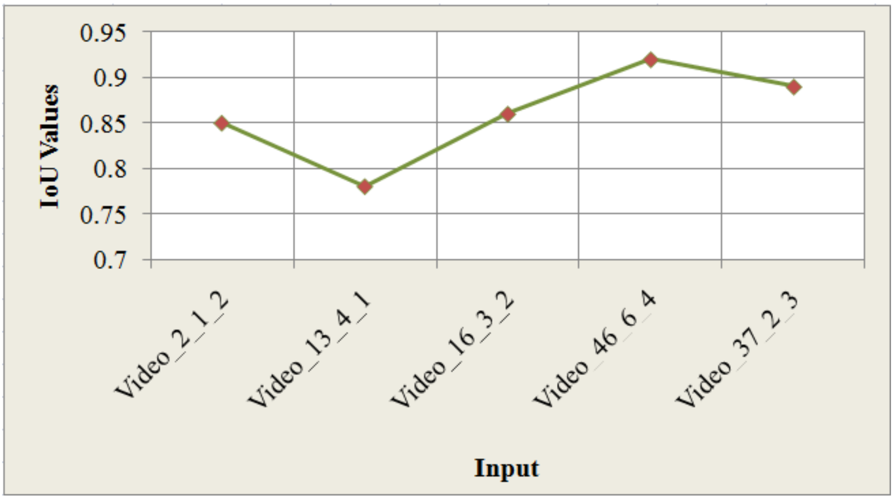

Figure 3: Analysis of IoU

Figure 3 presents IoU thresholds attained between oriented rectangles. It was observed from figure 3 that a total of 5 videos were used, while the IoU of those videos ranges from 0.78% to 0.92%. The analysis found that the quadrilateral results are more accurate than the rectangular results.

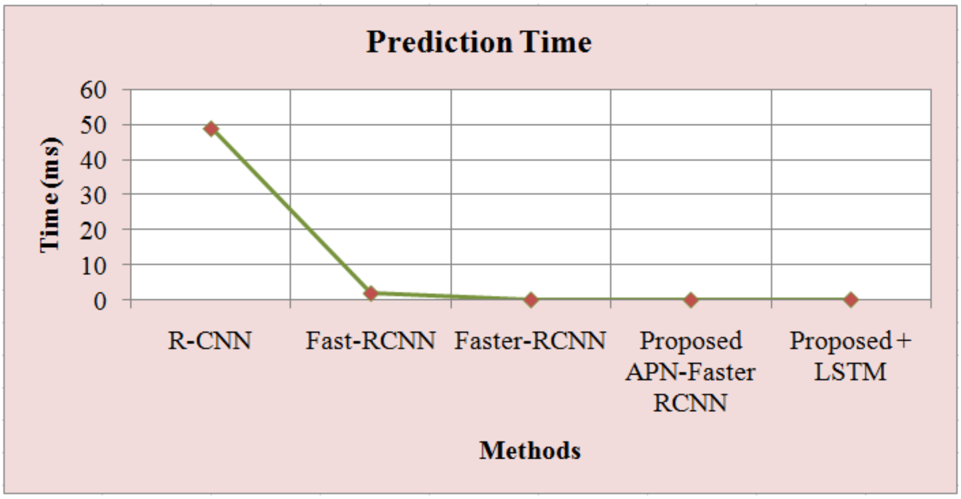

Figure 4: demonstrates the prediction time of the proposed and existing techniques

When analyzing the above figure 4 the time taken by the existing RCNN is 49 ms which is larger than other techniques. The detection time of the existing Fast R-CNN, and Faster R-CNN, is 2ms, and 0.2 ms whereas the proposed method takes 0.137 ms for detection. From the above analysis, it is clear that the proposed method proffers better performance than the existing methods.

Table 1: Analysis of Text Detection Accuracy

Models	**Accuracy (%)**
R-CNN	77.3
Fast-RCNN	86.03
Faster-RCNN	89
Proposed APN-Faster RCNN	91.2
Proposed + LSTM	88.3

Table 1 analyses the detection accuracy of the proposed and existing methods. The proposed method showed improved results than the existing methods where the existing RCNN, Fast-RCNN, and Faster RCNN attained an accuracy lower than the proposed method. The accuracy of the proposed method is improved by 13.9% compared to the existing R-CNN method. Thus, the proposed method is highly efficient and accurate in text detection.

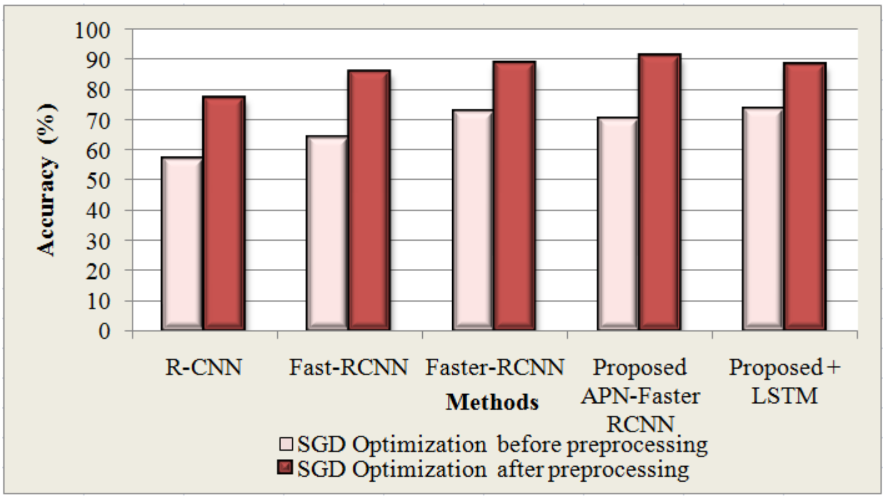

Figure 5: illustrates the accuracy of the proposed and existing methods

On analyzing the above figure 5, the accuracy attained by the proposed method is 91.2% whereas, before preprocessing it was 70.5%. This indicates that the preprocessing techniques used in the proposed model significantly improved the detection accuracy. Likewise, the existing methods that use preprocessing techniques attained accuracy lower than the proposed model. It clearly shows that the proposed method has higher accuracy than the existing methods for the detection of text.

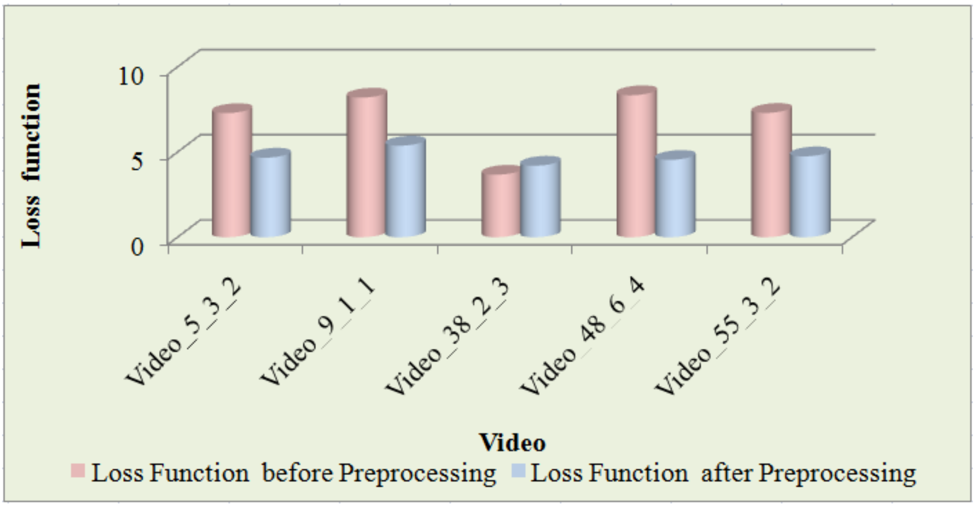

Figure 6: Analysis of Loss Rates

As shown in Figure 6, the proposed model attained reduced loss rates while using preprocessing. The loss rate of the proposed detection model varies between 4.2% and 5.38%, which ranges higher for the model that doesn't contain any preprocessing strategies. It is proven that the addition of preprocessing techniques can effectively reduce the prediction error indicating better detection accuracy of the model.

4.2 Comparative Analysis

In this section, the performance of the proposed model is weighted against the existing (Yu et al., 2021), (Xue et al., 2019), and (Huang, 2019) models based on precision, and recall.

Table 2: analysis of the performance of proposed APN-Faster RCNN with existing algorithms

Methods	Precision	Recall
(Yu et al., 2021)	75.08	52.28
(Xue et al., 2019)	64	71
(Huang, 2019)	84.50	87.60
Proposed Model	92.3	91.5

Table 2 shows the performance of the proposed method and the existing detection models with respect to some quality metrics. The above table 2 comparison clearly showed that the proposed method attained greater performance in terms of precision and Recall. The proposed method detects the text present in the video far better than the existing model developed in (Yu et al., 2021), (Xue et al., 2019), and (Huang, 2019).

V. CONCLUSION

In this paper, an efficient text detection method using FRCNN and Skeletonization is proposed. The main goal of this work is to detect texts in the video with higher accuracy. In the experiment, the performance of the proposed APN-FRCNN is compared with the existing approaches regarding quality metrics. Based on the performance analysis, the proposed FRCNN-APN achieves 91.9% accuracy and improved its precision and recall percentages by 28.3% and 39.22%. The analysis proves that the proposed scheme and its efficiency to detect texts are better than the existing methods. In the future, text detection with tracking and text recognition tasks are included in the proposed work for improving performance.

REFERENCES

[1]. Bhunia, A. K., Konwer, A., Bhunia, A. K., Bhowmick, A., Roy, P. P., & Pal, U. (2019). Script identification in natural scene image and video frames using an attention based Convolutional-LSTM network. Pattern Recognition, 85, 172–184. https://doi.org/10.1016/j.patcog.2018.07.034

[2]. Cai, Y., Liu, C., Wang, W., & Ye, Q. (2020). Towards Spatio-Temporal Video Scene Text Detection via Temporal Clustering. http://arxiv.org/abs/2011.09781

[3]. Cheng, Z., Lu, J., Zou, B., Qiao, L., Xu, Y., Pu, S., Niu, Y., Wu, F., & Zhou, S. (2021). FREE: A fast and robust end-to-end video text spotter. IEEE Transactions on Image Processing, 30, 822–837. https://doi.org/10.1109/TIP.2020.3038520

[4]. Feng, W., Yin, F., Zhang, X. Y., & Liu, C. L. (2021). Semantic-Aware Video Text Detection. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1, 1695–1705. https://doi.org/10.1109/CVPR46437.2021.00174

[5]. Guan, T., Gu, C., Lu, C., Tu, J., Feng, Q., Wu, K., & Guan, X. (2022). Industrial Scene Text Detection with Refined Feature-attentive Network. IEEE Transactions on Circuits and Systems for Video Technology, 14(8), 1–13. https://doi.org/10.1109/TCSVT.2022.3156390

[6]. Huang, X. (2019). Automatic video scene text detection based on saliency edge map. Multimedia Tools and Applications, 78(24), 34819–34838. https://doi.org/10.1007/s11042-019-08045-7

[7]. Jamil, A., Rasheed, J., & Bayram, B. (2019). Local statistical features for multilingual artificial text detection from video images. International Conference on Advance Technologies, Computer Engineering and Science (ICATCES), 2nd, 256–260. http://muh.karabuk.edu.tr/bilgisayar/icatces/proceeding_book_2019.pdf

[8]. Liu, X., Meng, G., & Pan, C. (2019). Scene text detection and recognition with advances in deep learning: a survey. International Journal on Document Analysis and Recognition. https://doi.org/10.1007/s10032-019-00320-5

[9]. Liu, Y., Jin, L., & Fang, C. (2020). Arbitrarily Shaped Scene Text Detection with a Mask Tightness Text Detector. IEEE Transactions on Image Processing, 29(c), 2918–2930. https://doi.org/10.1109/TIP.2019.2954218

[10]. Liu, Y., Jin, L., Zhang, S., Luo, C., & Zhang, S. (2019). Curved scene text detection via transverse and longitudinal sequence connection. Pattern Recognition, 90, 337–345. https://doi.org/10.1016/j.patcog.2019.02.002

[11]. Manita, S., Mansouri, S., Zrigui, M., & Berchech, S. (2021). Arabic text detection in news video using RetinaNet. Procedia Computer Science, 192, 796–803. https://doi.org/10.1016/j.procs.2021.08.082

[12]. Wang, L., Wang, Y., Shi, J., & Su, F. (2019). Video text detection by attentive spatiotemporal fusion of deep convolutional features. MM 2019 - Proceedings of the 27th ACM International Conference on Multimedia, 66–74. https://doi.org/10.1145/3343031.3350868

[13]. Wang, X., Feng, X., & Xia, Z. (2019). Scene video text tracking based on hybrid deep text detection and layout constraint. Neurocomputing, 363, 223–235. https://doi.org/10.1016/j.neucom.2019.05.101

[14]. Wang, Y., Wang, L., Su, F., & Shi, J. (2019). Video text detection with fully convolutional network and tracking. Proceedings - IEEE International Conference on Multimedia and Expo, 2019-July, 1738–1743. https://doi.org/10.1109/ICME.2019.00299

[15]. Xue, M., Shivakumara, P., Zhang, C., Lu, T., & Pal, U. (2019). Curved text detection in blurred/non-blurred video/scene images. Multimedia Tools and Applications, 78(18), 25629–25653. https://doi.org/10.1007/s11042-019-7721-2

[16]. Yu, H., Huang, Y., Pi, L., Zhang, C., Li, X., & Wang, L. (2021). End-to-end video text detection with online tracking. Pattern Recognition, 113, 107791. https://doi.org/10.1016/j.patcog.2020.107791

[17]. Zhang, S. X., Zhu, X., Hou, J. B., Liu, C., Yang, C., Wang, H., & Yin, X. C. (2020). Deep Relational Reasoning Graph Network for Arbitrary Shape Text Detection. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 9696–9705. https://doi.org/10.1109/CVPR42600.2020.00972

[18]. Zhu, Y., & Du, J. (2021). TextMountain: Accurate scene text detection via instance segmentation. Pattern Recognition, 110. https://doi.org/10.1016/j.patcog.2020.107336

Mortha Manasa Devi, et. al. “A Novel Hybrid Architecture For Accurate Text Detection In Video With Text Skeletonization And Reinforced Frcnn.” *IOSR Journal of Computer Engineering (IOSR-JCE)*, 24(6), 2022, pp. 49-60.