*Dissertation on*

# "Multilingual Subtitle Detection and Removal Using Video Inpainting"

*Submitted in partial fulfilment of the requirements for the award of degree of*

**Bachelor of Technology**
**in**
**Computer Science & Engineering**

**UE19CS390B – Capstone Project Phase - 2**

***Submitted by:***


| | |
|---|---|
| **Ramya C** | **PES1UG19CS379** |
| **Ratna Bojja** | **PES1UG19CS381** |
| **Rishab S** | **PES1UG19CS386** |
| **Sahana B Manjunath** | **PES1UG19CS411** |


*Under the guidance of*


**Prof. V R Badri Prasad**
Associate Professor
PES University


**August - December 2022**


**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**
FACULTY OF ENGINEERING
**PES UNIVERSITY**
(Established under Karnataka Act No. 16 of 2013)


**PES UNIVERSITY**

# PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
100 Feet Ring Road, Bengaluru – 560 085, Karnataka, India

**FACULTY OF ENGINEERING**

# CERTIFICATE

*This is to certify that the dissertation entitled*

## 'Multilingual Subtitle Detection and Removal Using Video Inpainting'

*is a bonafide work carried out by*

| | |
|---|---|
| **Ramya C** | **PES1UG19CS379** |
| **Ratna Bojja** | **PES1UG19CS381** |
| **Rishab S** | **PES1UG19CS386** |
| **Sahana B Manjunath** | **PES1UG19CS411** |

in partial fulfilment for the completion of seventh semester Capstone Project Phase - 2 (UE19CS390B) in the Program of Study - Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period August - December 2022. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 7$^{th}$ semester academic requirements in respect of project work.

| | | |
|---|---|---|
| Signature<br>Prof. V R Badri Prasad<br>Associate Professor | Signature<br>Dr. Shylaja S.S.<br>Chairperson | Signature<br>Dr. B.K. Keshavan<br>Dean of Faculty |

**External Viva**

| **Name of the Examiners** | **Signature with Date** |
|---|---|
| **1.** _____ | _____ |
| **2.** _____ | _____ |

# DECLARATION

We hereby declare that the Capstone Project Phase - 2 entitled **"Multilingual Subtitle Detection and Removal Using Video Inpainting"** has been carried out by us under the guidance of Prof. V R Badri Prasad, Associate Professor and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester August - December 2022. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

| | |
|---|---|
| **PES1UG19CS379** | **Ramya C** |
| **PES1UG19CS381** | **Ratna Bojja** |
| **PES1UG19CS386** | **Rishab S** |
| **PES1UG19CS411** | **Sahana B Manjunath** |

# ACKNOWLEDGEMENT

We would like to express our gratitude to Prof. V R Badri Prasad, Department of Computer Science and Engineering, PES University, for his/her continuous guidance, assistance, and encouragement throughout the development of this UE19CS390B - Capstone Project Phase – 2.

We are grateful to the project coordinator, Prof. Mahesh H.B., for organizing, managing, and helping with the entire process.

We take this opportunity to thank Dr. Shylaja S.S. Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support we have received from the department. We would like to thank Dr. B.K. Keshavan, Dean of Faculty, PES University for his help.

We are deeply grateful to Dr. M.R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J., Vice-Chancellor, PES University for providing us various opportunities and enlightenment every step of the way.

Finally, this project could not have been completed without the continual support and encouragement we have received from our family and friends

# ABSTRACT

Multilingual Subtitle Detection and Removal using Video Inpainting project aims at building a framework to remove subtitles that are unnecessary to the viewer using deep learning techniques. This can be achieved with a pipeline of two main modules, subtitle text detection and removal of the subtitles using video inpainting technique.

Starting with the high contrast property of subtitles, they are separated from the background text using the colour segmentation technique. To remove the remains of other high-contrast background noise for detecting the subtitle region effectively, a text detection model is employed that detects the subtitle region across all the frames in the video. This region is split into lower dimension sub-parts for faster and more efficient inpainting, also preserving the resolution of the original video. The inpainted frames along with audio are merged back into a video without subtitles.

# TABLE OF CONTENTS


| Chapter No. | Title | Page No. |
|---|---|---|
| **1.** | **INTRODUCTION** | **01** |
| | 1.1 Pre-processing | **02** |
| | 1.2 Text Detection | **02** |
| | 1.3 Video Inpainting | **02** |
| | 1.4 Construction of video from frames | **02** |
| **2.** | **PROBLEM STATEMENT** | **03** |
| | 1.1 Defining Problem Statement | **03** |
| | 1.2 Background of the problem statement | **03** |
| **3.** | **LITERATURE REVIEW** | **04** |
| | 3.1 Text Detection Module | **05** |
| | 3.2 Image Inpainting Module | **14** |
| | 3.3 Video Inpainting Module | **21** |
| | 3.4 Complete pipeline | **26** |
| **4.** | **DATA** | **28** |
| | 4.1 Overview | **28** |
| | 4.2 Datasets | **28** |
| | 4.2.1 Video Dataset | **28** |
| | 4.2.2 Text Detection Dataset | **29** |
| | 4.3 Statistics | **30** |
| **5.** | **METHODOLOGY** | **32** |
| | 5.1 Pre-processing | **32** |
| | 5.1.1 Splitting video to video frames | **32** |
| | 5.1.2 Image segmentation | **33** |

5.2 Text Detection Module 34
5.2.1 Dataset 34
5.2.2 Architecture 35
5.2.2.1 Code 36
5.2.3 Training 36
5.2.4 Output and Split Region 36
5.3 Video Inpainting 38
5.3.1 Dataset 38
5.3.2 Architecture 38
5.3.2.1 Code 39
5.3.3 Output 40
5.4 Construction of Video from Image Frames 40
5.4.1 Rejoin the Frames 40
5.4.2 Add Audio 40
**6. RESULTS AND DISCUSSION 42**
6.1 Text Detection 42
6.1.1 Intersection over Union (IoU) 42
6.1.2 Precision 42
6.1.3 Recall 43
6.1.4 F1-Score 43
6.2 Video Inpainting 45
6.2.1 Peak Signal to Noise Ratio (PSNR) 45
6.2.2 Structural Similarity Index Measure (SSIM) 45
**7. CONCLUSION AND FUTURE WORK 47**
**REFERENCES/BIBLIOGRAPHY 48**
**APPENDIX A DEFINITIONS, ACRONYMS AND ABBREVIATIONS 51**

# LIST OF FIGURES


| Figure No. | Title | Page No. |
|---|---|---|
| **01** | CTPN Architecture | **06** |
| **02** | Conversion of Video to VideoFrames | **07** |
| **03** | Conversion of Video to Video Frames for Text Detection | **08** |
| **04** | Sample images showing the bounding box results after each sub-step to generate the bounding box proposals | **11** |
| **05** | Images showing output (black for text and white for non-text) results of the refinement module. | **11** |
| **06** | Implementation of doubling strategy | **11** |
| **07** | Sample images showing the refinement module results | **12** |
| **08** | Conversion of video to video frames in SMPM | **15** |
| **09** | LaMa architecture | **16** |
| **10** | EdgeConnect architecture | **17** |
| **11** | Flow of work | **18** |
| **12** | Edge Connect | **18** |
| **13** | Patch Match | **19** |
| **14** | Deep Image Prior | **19** |
| **15** | Evaluation results | **20** |
| **16** | SSIM and PSNR results | **20** |

| | | |
|---|---|---|
| **17** | Architecture of Joint Spatial-Temporal Transformation for Video Inpainting | **22** |
| **18** | Architecture of Copy-Paste Network | **23** |
| **19** | Conversion of video to video frames by multi-patch based attention | **24** |
| **20** | Flow-Guided Architecture | **25** |
| **21** | Overview of the Proposed Model | **26** |
| **22** | CTPN with EdgeConnect architecture | **27** |
| **23** | High level view of the pipeline | **32** |
| **24** | Splitting video to video frames | **33** |
| **25** | Frame Segmentation | **33** |
| **26** | CTPN Architecture with Bi-GRU of Text Detection Module | **35** |
| **27** | Detecting text in the segmented binary image using CTPN model | **37** |
| **28** | Generating the complete bounding box | **37** |
| **29** | Splitting the subtitle region to 240 x 432 dimensions | **38** |
| **30** | E2FGVI Architecture | **38** |
| **31** | Inpainted Image Output | **40** |
| **32** | Inpainted video | **41** |
| **33** | Model Accuracy | **44** |
| **34** | Model Loss | **44** |

# LIST OF TABLES


| Table No. | Title | Page No. |
|---|---|---|
| **01** | Evaluation results of MSER on different datasets | **09** |
| **02** | Precision, Recall and F1 Score were calculated against different datasets | **10** |
| **03** | Results obtained after the refinement module on different datasets with and without using doubling strategy | **12** |
| **04** | Performance comparison between the proposed method and some state-of-the-art methods for scene text detection and localization on standard datasets. | **13** |
| **05** | Dataset General Specifications | **30** |
| **06** | Video Dataset Specifications | **30** |
| **07** | Text Detection Dataset Specification | **31** |
| **08** | Train accuracies for different epochs | **43** |
| **09** | Text Detection Model Accuracy | **45** |
| **10** | Video Inpainting Model Accuracy | **46** |

# CHAPTER 1

# INTRODUCTION

Subtitles are the artificial text that is either overlaid (soft subtitles) or embedded (hard subtitles) in a video for people to understand the content in the video better. In many of the videos, subtitles exist in the form of hardcoded/embedded fashion. They become a merit for people who cannot understand the audio running in the background or if someone has hearing impairment. Although, these language-specific subtitles become a demerit when the viewer does not understand the language of the subtitle. Therefore removal of these subtitles helps the viewer to watch the subtitle-free content or overlay another set of subtitles, of preferred language, on the video for better understanding.

The removal of the embedded subtitles from the video leads to degradation of the pixels in the subtitle region, due to which the video quality is drastically reduced. Therefore, the major aspects to be taken into consideration while removing the subtitle text is to preserve the quality of the video and avoid removal of scene text. To generate a subtitle-free video, a text detection model is built to detect the subtitle text in the video. To restore the degraded/subtitle text pixels, one way is to perform image inpainting on each image frame. This results in a temporally inconsistent video. Therefore, a video inpainting model is employed to fill the subtitle region with necessary content. Our contributions include building an end to end framework for detecting multilingual subtitles in the video and removing them using video inpainting technique. The various stages to achieve this goal are specified as preprocessing, text detection, video inpainting and postprocessing.

## 1.1 Pre-Processing

In the Pre-Processing stage, we use OpenCV to convert the video into image frames and also extract audio from it. They are named in serial order to avoid misinterpretation of the frames. We also use MoviePy to extract the audio clip from the video. The image frames are further transformed from BGR to HSI model. With thresholding based on the intensity and saturation thresholds, the image is converted to a binary format. Subtitles are the high intensity/contrast components in the videos.

## 1.2 Text Detection

For detecting only subtitle text, excluding the scene text because it is necessary content in the video, CTPN (Connectionist Text Proposal Network) model which was trained on an in-house dataset of 3.5k binary images (segmented images by thresholding) with subtitle text, background noise and scene text. The output of this model is a bounding box around the subtitle region.

## 1.3 Video Inpainting

The second stage, the output of the text detection model, frame with subtitle mask will be taken as input to End to End Framework for Flow Guided Video Inpainting, After obtaining the binary mask from text detection. we split it into segments and that is sequentially passed along with the input video. Finally, the output of this model will be the inpainted subtitle-free frame.

## 1.4 Construction of the video from frames

After inpainting the subtitle region in all the frames we need to construct the video back from the frames and sync the audio to get the final output video. OpenCV and MoviePy libraries are used to generate this output.

# CHAPTER 2

# PROBLEM STATEMENT

## 2.1 Defining Problem Statement

"To design a pipeline for converting a video with hard-coded multilingual subtitles to a subtitle-free video."

## 2.2 Background for the problem statement

Videos are generated from multiple sources across the world. To understand the content of the videos without knowing the language, 'subtitles' are essential.

But there are cases in which the subtitles are irrelevant to the user. For example, the user might not understand the language of the subtitles, therefore the subtitles become a source of distraction to the user, and she/he would rather prefer to remove these subtitles and maybe further embed the subtitles of the language she/he understands. Removing the subtitles is possible in the case of closed captions (subtitles that are generated by transcript and laid over the video, can be seen as a feature on YouTube). Closed Captions belong to the class 'Soft subtitles', these can be easily removed from the video without any damage.

'Hard subtitles' are completely embedded in the video and any attempt to remove these subtitles would create a void in the subtitle region. There are many videos on the web with hard-coded subtitles (including YouTube) that have embedded subtitles, and the closed-captions option is disabled for these videos, so the subtitles in the video cannot be removed. To remove the hard-coded subtitles, 'Subtitle Text Detection and Inpainting' (filling the void with the right information) has to be performed to get a subtitle-free video.

# CHAPTER 3

# LITERATURE SURVEY

The Chapter Literature Survey deals with the findings in the Removal of Subtitles through Video Inpainting which has been useful to identify the novelty for the proposed problem statement and the flow of the project. A comprehensive methodical review is performed based on various aspects that are dependent or useful for the project.

The aspects consists of:

- Various analysis of research papers and the products relevant to the problem statement.
- The information provided through various search engines that is suitable to the chosen problem statement.
- Many detailed designs of products or prototypes which are appropriate to the picked up problem statement.
- Analysis of various methodologies that have been discussed in various research papers.
- Implemented techniques and validation of their evaluation metrics and performance analysis.

There are some good sequential models for Text Detection with higher accuracy and for Video Inpainting which preferably has used 3D CNNs and transformers to achieve good results.

The further section deals with the understanding and analysis of various models (Text Detection and Video Inpainting) which includes the scenarios and substantial findings to design our project.

## 3.1 Text Detection Module

A detailed research review on the techniques and methodologies used to detect text in an image.

**Zhi Tian et al. [1],** proposed a novel deep learning architecture to detect scene text. The Connectionist Text Proposal Network (CTPN) architecture comprises an object detection model (VGG-16) which extracts a convolutional feature map of the input image. Further, a fine-scale text proposal method is introduced to predict the text regions by sliding a single window through the feature map. The fine-scale text proposal technique employs a vertical-anchor mechanism which proposes regions with fixed width and varying height boxes called anchors, and these regions are filtered based on their text/non-text scores.

The regions (coordinates in the image) with a text score higher than 0.7 threshold are then passed through the Bi-LSTM model which generates continuous text proposal regions to build a text line. The CTPN model outputs coordinates of bounding boxes around each text line in the image, and these bounding boxes are constructed by joining the close text proposals together and side refinement, to not leave out any text in the left and right ends of the text line.

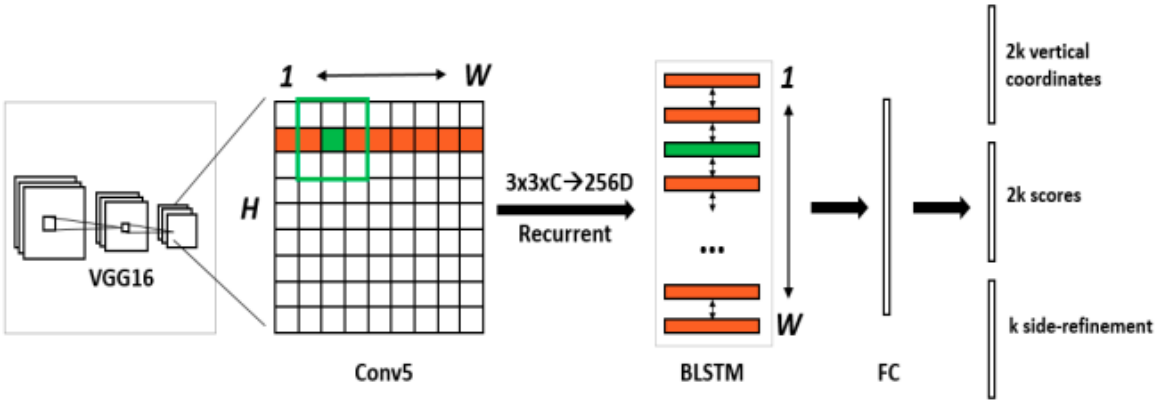


Figure 01: CTPN Architecture

**A.Jamil et al. [2]** explained the use of statistical features to detect multilingual artificial text from image frames with complex backgrounds. The dataset consists of image frames extracted from videos from multiple sources which have bilingual horizontal text , English and Urdu.

In the proposed methodology, the input image is passed through multiple phases of generating low-level features. The first phase involves identifying the text using local entropy and gradient difference between the text and the background. Next morphological operations are performed to join the edges of each character in order to identify the text line. Then the text lines are drawn with horizontal projections.

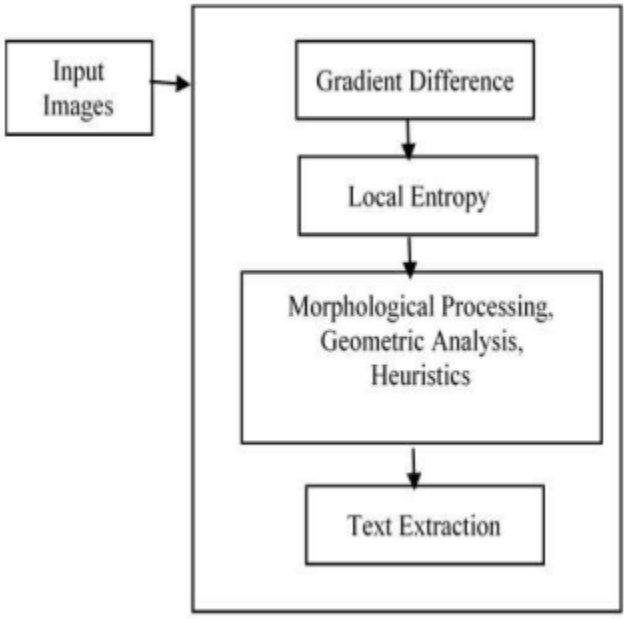


Figure 02: Conversion of video to video frames

**Minghui Liao et al. [3]** proposed a novel method to localize scene text and recognize it with high speed and accuracy compared to all the state of the art models. The output is the coordinates of the bounding boxes drawn around each word in the text. The image with scene text is passed through a fully convolutional neural network only once for text localization. Further, the CRNN model is employed for scene text recognition and also the semantics of the text helps the detection to be more accurate. TextBoxes model takes around 0.09 seconds per image to detect the text.

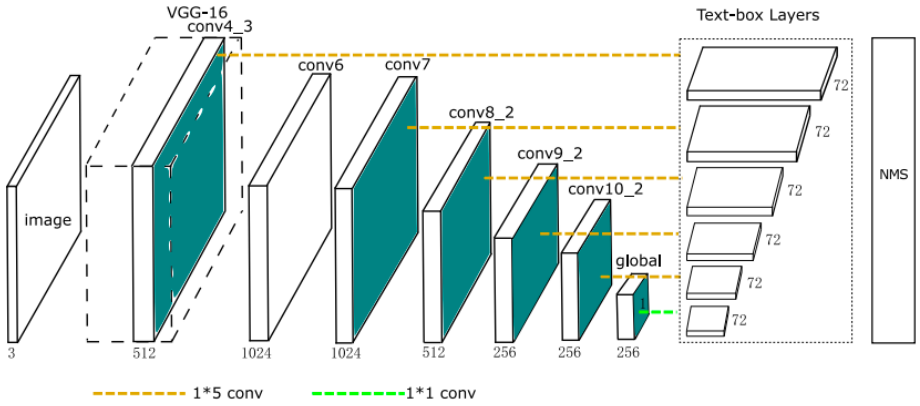


Figure 03: Conversion of video to video frames for Text Detection

**Mikhail Zarechensky et al. [4]** conversed about different algorithms related to text detection and probed them for different languages. Mainly connected components-based methods are considered.

The paper discusses the Maximally Stable Extremal Region algorithm, where the input image is binarized with a threshold iterating from 0 to 255, for every extremal region, number of successive images detected in sequence where this region stays the same. We can choose these regions called Maximally Stable Extremal Regions or MSER.

The second algorithm discussed is from Multi-script text extraction from natural scenes, using the ICDAR 2013 dataset. The algorithm first applied MSER and then filtered the regions based on intensity, stroke width variance, aspect ratio, number of holes, the colour of the outer boundary, and gradient magnitude.

The test datasets (MSRA TD-500, ICDAR 2011 AND ICDAR 2013) are used to determine the precision, f-measure and recall of the models.

Table 1: Results on ICDAR 2011 dataset

| Methods | recall | precision | $f$-measure |
|---|---|---|---|
| Yin et al. | 0.68 | 0.86 | 0.76 |
| Chen et al. | 0.60 | 0.73 | 0.66 |

Table 2: Results on MSRA-TD500 dataset

| Methods | recall | precision | $f$-measure |
|---|---|---|---|
| Yin et al. | 0.21 | 0.517 | 0.335 |

Table 3: Results on ICDAR 2013 dataset

| Methods | recall | precision | $f$-measure |
|---|---|---|---|
| Yin et al. | 0.42 | 0.64 | 0.51 |

Table 4: Results on special dataset

| Methods | recall | precision | $f$-measure |
|---|---|---|---|
| Yin et al. | 0.079 | 0.577 | 0.109 |
| Chen et al. | 0.071 | 0.427 | 0.299 |

Table 01: Evaluation results of MSER on different datasets

**Youngmin Baek et al. [5]** suggested architecture in the subsequent research uses both the provided character level annotations for synthetic images and the approximate character-level ground-truths for real images obtained by learning intermediate models to address the deficiencies of each character level annotation.

The model has a fully convoluted network architecture that has been pre-trained and is based on the batch normalisation model VGG 16. This serves as the main support structure for the building. During the decoding phase, this model also features skip connections, which is probably due to the U-net architecture. The main application of this is to cluster low level features. The region score and affinity score are the two channels that make up the final result.

**Shaswata Saha et al.[6],** It was suggested to use MSER, SWT, and GAN to refine candidate scene text areas and a CNN-based classifier to identify languages. A multilingual Indic scene text dataset created in-house as well as datasets from KAIST, COCO, CTW1500, CSVI, and ICDAR are used to test this design.

With a bounding box indicating the potential text area utilising MSER, the initial prospective scene text regions are suggested. To decrease the non-text components, SWT is applied to the acquired pictures. Final candidate scene text suggestions are created by considering the characters in these pictures that have few spaces between them to be a single word and contained within a rectangular box. These suggestions are further refined using an image-to-image translation framework to only include textual components. Augmented images ensure that a text region is completely covered. Language detection is accomplished using a CNN architecture.

The results can be seen in the following tables:

| Dataset | No. of Images | Precision | Recall | F1 Score |
|---|---|---|---|---|
| KAIST | 600 | 0.641 | 0.730 | 0.683 |
| ICDAR 2003 | 259 | 0.553 | 0.676 | 0.608 |
| ICDAR 2011 | 485 | 0.587 | 0.685 | 0.575 |
| ICDAR 2013 | 229 | 0.554 | 0.672 | 0.540 |
| ICDAR 2015 | 229 | 0.532 | 0.672 | 0.517 |
| ICDAR 2019 | 10000 | 0.555 | 0.672 | 0.567 |
| COCO Text | 2700 | 0.265 | 0.826 | 0.368 |
| CTW 1500 | 1500 | 0.582 | 0.860 | 0.682 |
| In-house | 100 | 0.453 | 0.551 | 0.497 |

Table 02: Precision, Recall and F1 Score were calculated against different datasets


Figure 04: Sample images showing the bounding box results after each sub-step to generate the bounding box proposals

Figure 05: Images showing output (black for text and white for non-text) results of the refinement module.


Figure 06: Implementation of doubling strategy


Figure 07: Sample images showing the refinement module results.

| Dataset | Doubling Strategy | No. of Images | Precision | Recall | F1 Score |
|---|---|---|---|---|---|
| **KAIST** | Yes | 600 | 0.854 | **0.846** | **0.845** |
| **KAIST** | No | 600 | **0.929** | 0.309 | 0.429 |
| **ICDAR** | Yes | 1000 | 0.744 | **0.802** | **0.753** |
| **ICDAR** | No | 1000 | **0.871** | 0.231 | 0.318 |
| **In-house** | Yes | 100 | 0.601 | **0.579** | **0.582** |
| **In-house** | No | 100 | **0.625** | 0.192 | 0.262 |

Table 03: Results obtained after the refinement module on different datasets with and without using doubling strategy

| Dataset | Method | Precision (%) | Recall (%) | F1 score (%) |
|---|---|---|---|---|
| **ICDAR 2003** | [33] | 68.8 | 66 | 66 |
| | Proposed | **71.1** | **83.2** | **76.7** |
| **ICDAR 2011** | [35] | 89.2 | 62.3 | 73.3 |
| | [8] | **91.5** | 74.8 | 82.3 |
| | Proposed | 89.6 | **91** | **90.3** |
| **ICDAR 2013** | [34] | 88.9 | 80.2 | **84.3** |
| | [8] | **92** | 75.5 | 83 |
| | Proposed | 78.9 | **86.1** | 82.3 |
| **ICDAR 2015** | [34] | 72.3 | 58.7 | 64.8 |
| | [8] | 87 | 77 | 82 |
| | [9] | 82 | 80 | 81 |
| | Proposed | **88.4** | **88.2** | **88.3** |
| **KAIST** | [12] | 85 | **90** | 87.4 |
| | Proposed | **85.4** | 84.6 | 84.5 |
| **COCO Text** | [34] | 43.2 | 27.1 | 33.3 |
| | Textboxes++ [16] | **60.9** | 56.7 | 58.7 |
| | Proposed | 59.8 | **83** | **69.5** |
| **ICDAR 2019** | Tencent-DPPR [24] | **87.52** | 80.1 | 83.6 |
| | Proposed | 82.1 | **80.5** | 81.3 |
| **CTW1500** | [18] | **78.7** | 76.1 | 77.4 |
| | Proposed | 77 | **76.5** | 76.8 |

Table 04: Performance comparison between the proposed method and some state-of-the-art methods for scene text detection and localization on standard datasets

The performance of the proposed system needs to be improved, and tests are needed to see how well it stands up to increasingly complex images.

## 3.2 Image Inpainting Module

A detailed research review of the techniques and methodologies used to inpaint the holes in an image frame using the image inpainting model.

Image Inpainting is a long-studied task in the field of computer vision and image processing. The field of research has been going on since the 2000s. The primary goal of image inpainting was to remove the damaged portions of a vintage picture by completing the remaining image from the global pixels of the image. The primary techniques used include the analysis and utility of pixel properties in the spatial and frequency domain.

**Mohammad Khodadadi et al. [7]** proposed a new algorithm for text detection, text extraction and text inpainting that includes three stages, extract blocks of text, precise text characters extraction and last, an inpainting algorithm to erase and fill the regions with the original texture of the background.

To detect blocks in the image frame they used the stroke filter using the intensity of colour in RGB channels, which creates a solid colour outline on an object. Post the stroke filter process, regions with high values are retained and the left out regions are set to zero, based on the threshold calculated using local and global mean and variance. Test edges density filter is used to fill out the gaps in between. To extract the text from text blocks histogram is built for each colour channel for the text and as well as the background areas to detect the text character in the candidate regions.

The introduced inpainting algorithm is based on texture synthesis and matching. This algorithm repairs the damaged pixel by obtaining the best matching pixel. The damaged pixel is selected first based on the least damaged pixels in its 8*8 neighbours.

The best matching pixel is found based on the similarity which is calculated using Sum Squared Difference (SSD).

**Zhengmi Tang et al. [8]** proposed a network containing a Stroke Mask Prediction Module (SMPM) that filters out the text stroke as a relatively small region from the clipped text region to preserve background information to a greater extent for obtaining better inpainting output from Background Inpainting Module (BIPM).

They introduce a word-level two-stage scene-text-erasing network that works on clipped text images. They have adopted an encoder-decoder Fully Convolutional Network (FCN) to extract feature maps from the residual blocks (text) and feature maps of the background, by which text and non-text regions are differentiated. In BIPM architecture, Partial Convolutional Layers are used to generate clearer background images and cover up artefacts and text ghosts, and Skip Connection and Self-attention Blocks are used to obtain more features from outlying non-text regions.

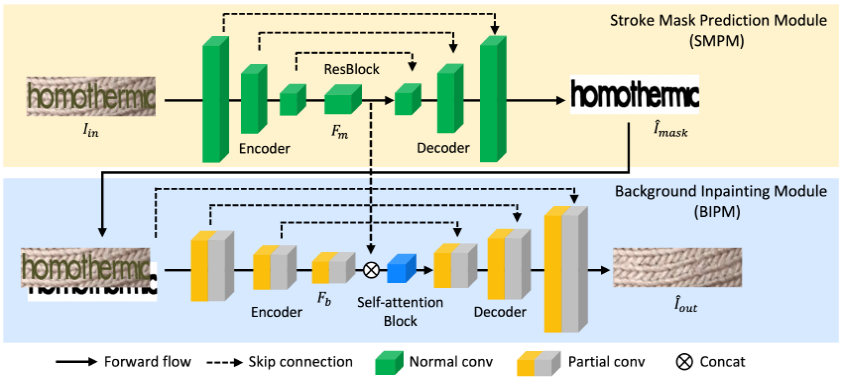


Figure 08: Conversion of video to video frames in SMPM

**Roman Suvorov et al. [9]** have responsive fields of these convolutions that span the entire image. The suggested technique can effectively paint wide areas and performs well with a wide variety of images, even those with intricate repeated structures. Despite the model being trained on photos with lesser resolutions like 256X256, the method also works for much greater resolutions.

LaMa is based on a feed-forward inpainting network that is similar to ResNet and makes extensive use of the recently proposed fast Fourier convolutions, which have a significant impactful receptive field and can handle recurrent structures. It also uses a multi-component loss that combines adversarial loss and a high receptive field perceptual loss, as well as a training-time for large mask generation process.

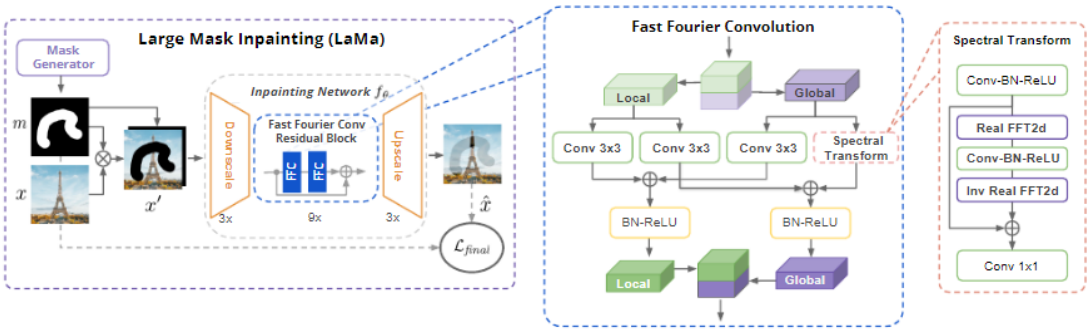


Figure 09 : LaMa architecture

**Kamyar Nazeri et al. [10]** proposed a model that works in two massive phases.The first one is edge generation, wherein the model starts to hallucinate pixels or, more precisely, edges in the parts of the screen that seem to be absent. After the hallucinated edges are obtained, the image completion network helps in the estimation of the RGB intensity in those missing regions thus generating an in painted image as the output.

The edge generator uses a generator discriminator pair. The edge generator uses the mask and the image provided input and then converts the input image to grayscale output would be the hallucinated image.They generator has not been changed much but the discriminator used is a PatchGAN of size 70X70.

For the image completion network, the input is the hallucinated image obtained as output from the edge connector. This runs through a dilated convolutional network that gives out the input image along with a binary mask filling up the edges. These are again passed to a dilated convolutional network that fills the binary mask based on the spatial information available from the input image and delivers the inpainted image as an output.

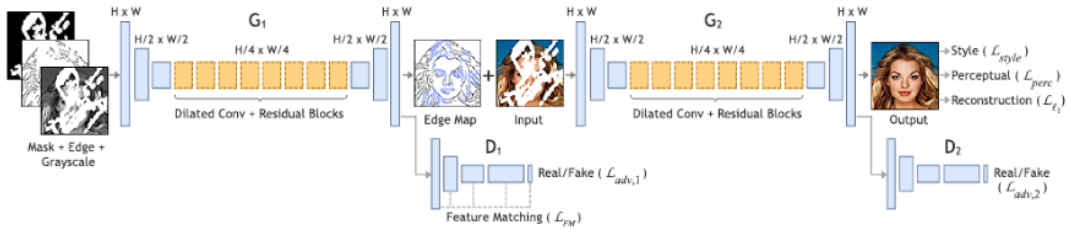


Figure 10: EdgeConnect architecture

**Guilin Liu, Fitsum A et al. [11]** used Partial Convolutions wherein the convolutions are the masks that are resized and further renormalized such that it is conditioned only to valid and liable pixels.

The model generates these masks and renormalises them from the process of segmentation aware convolutions.The primary focus is on the holes or the incomplete parts of the image. This renormalised mask is passed through U-Net architecture for image inpainting. Since only the corrupted parts of the image are considered for inpainting, the computational cost hence reduces.

**Dr. A. Pasumpon Pandian [12]**, uses a hybridised image in-painting approach that combines edge connect, patch match, and the DIP prior for image in-painting to produce high-quality, high-resolution images (image restoration and improvement).

The suggested design makes use of and incorporates the edge connect, patch match, and deep image previous in order to improve the image quality and resolution.

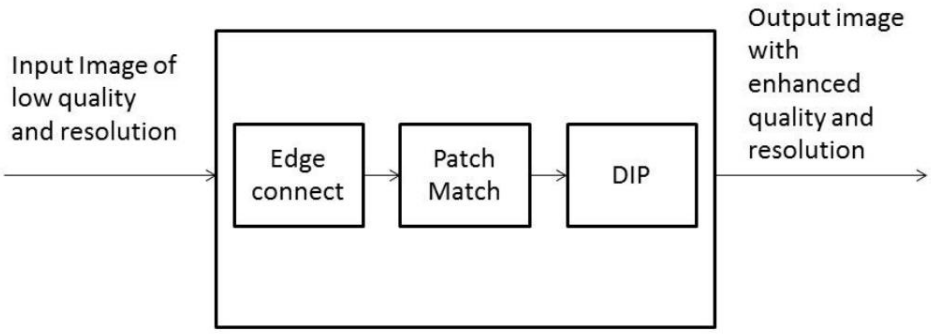


Figure 11: Flow of work

Edge Connect is used to find important edges of missed areas in photographs using the edge generator.

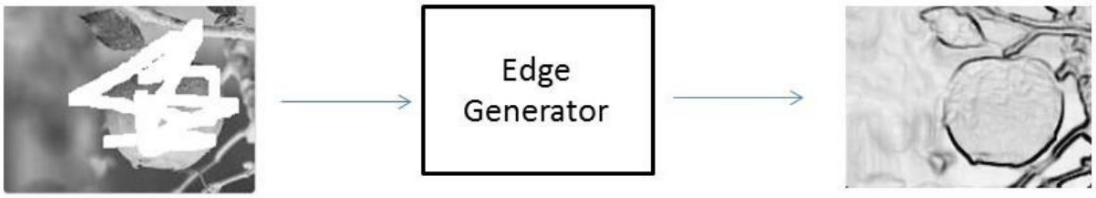


Figure 12: Edge Connect

A programme called Patch Match is used to find picture patch matches rapidly. To find the closest and most pertinent matches among the image patches, it also makes use of random sampling and natural coherence.

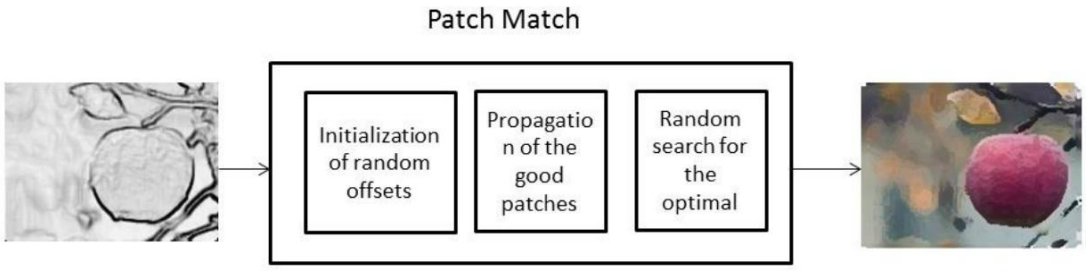


Figure 13: Patch Match

DIP is used to improve the image quality and resolution even further.

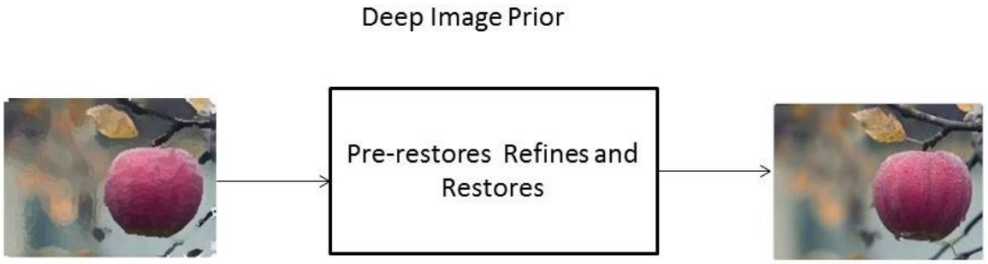


Figure 14: Deep Image Prior

SSIM and PSNR are used as evaluation metrics.


Figure 15: Evaluation results

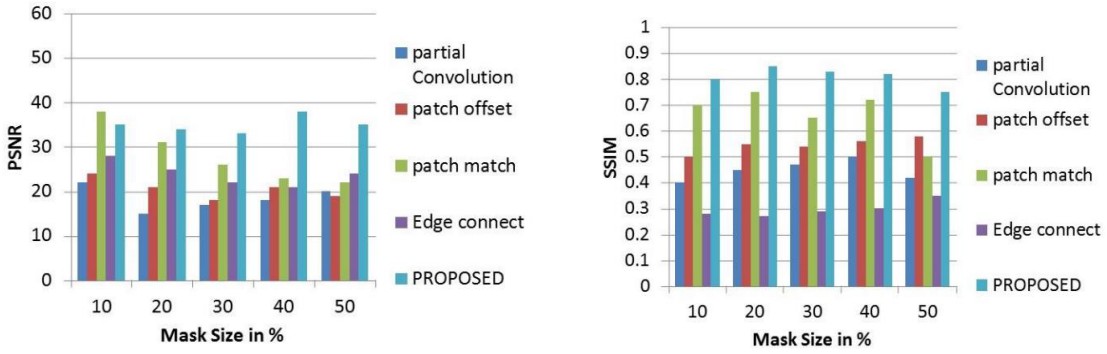


Figure 16: SSIM and PSNR results

Future Work of the paper is Plan to expand their work to detect text in multiple Indian languages.

## 3.3 Video Inpainting Module

A detailed research review on the techniques and methodologies used to inpaint the holes in an image frame using video inpainting models.

Apart from image inpainting, video inpainting is also used in object removal. There are several strategic algorithms and methodologies that were used to remove the static objects of the image as a part of image inpainting. With a considerable proportion of object tracking and image completion, the task was further extended to video inpainting. There are many old videos with spikes and grey vertical lines. With precision, video inpainting has been successful to achieve that as well.

**Zhen Li1 et al. [13]** proposed a model where primarily, they have used a context encoder whose major function 1s to encoding the corrupted frames into lower resolution frames which has a better computational efficiency at succeeding processing. Further they extract and follow through the optical flow between local neighbors through the flow completion module. Third, the finished optical flow aids the feature extracted from local neighbors and accomplishes feature alignment and bidirectional propagation. Fourth, multi-layer temporal focal transformers perform content hallucination by combining propagated local neighboring features with non-local reference features. Finally, a decoder up-scales the filled features and reconstructs them to a final video sequence.

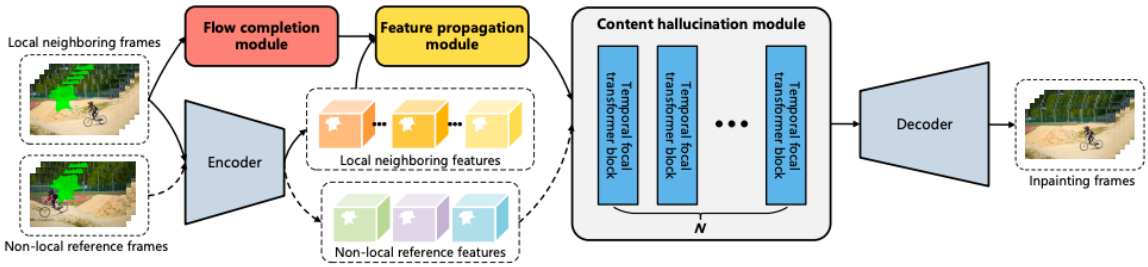


Figure 17: Architecture of Joint Spatial-Temporal Transformation for Video Inpainting

**Sungho Lee et al. [14]** proposed a method to copy similar pixels in the neighboring frames and paste it in the voids of the target frame. The authors proposed a novel Deep Neural Network to perform video inpainting. The network also includes an alignment network which computes the affine transformations between frames to align similar frames together. And, for inpainting a particular frame in the video effectively, information from distant frames is also considered.

The inpainting speed of a frame is faster than any other state of art methods. The affine transformations provide a large space of temporal information as more frames, distant frames, are referred for inpainting a particular frame. Hence the temporal information is conserved better than the optical flow methods as well.

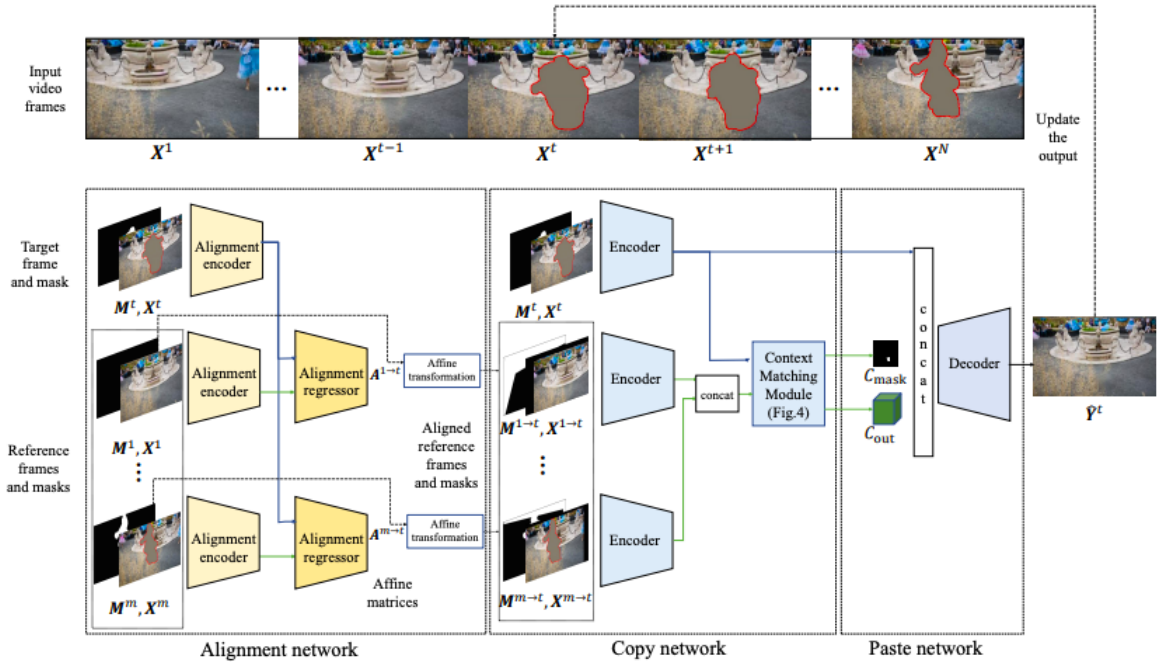


Figure 18: Architecture of Copy-Paste network

**Yanhong Zeng et al. [15]** proposed a novel transformer network for inpainting a video using multi-patch based attention mechanism. The main motive is to inpaint videos with complex motions. To fill a missing region in the image frame, the spatio-temporal transformer searches for similar patches in the neighborhood and distant frames using the multi-head module along temporal and spatial dimensions. Patches of different scales are captured which are further used for the process of inpainting.

The model also performs optimization on the temporal relationship between frames. Training time is also reduced as the spatio-temporal transformers can be simultaneously trained using multiple attention heads.

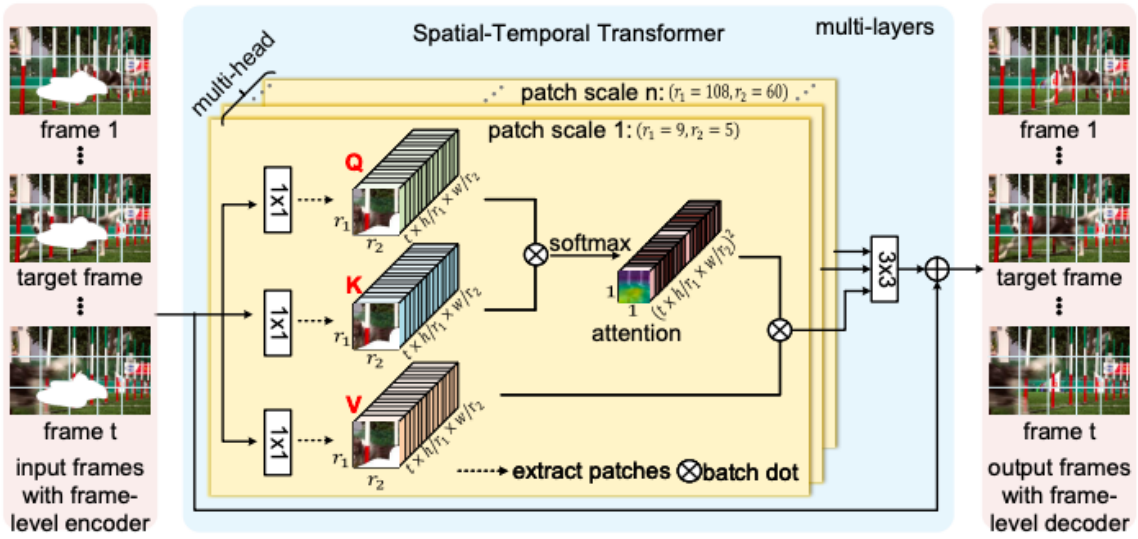


Figure 19: Conversion of video to video frames by multi-patch based attention

**Kaidong Zhang et al. [16]** proposed a flow completion network being used to repair corrupted flows using temporal features and distorted frames using spatial features. The model's leverage is the motion discrepancy that originates during object flow. They introduce window partition techniques for both temporal and spatial transformers to enhance the model's efficiency.

The steps in the suggested technique are listed below. The contaminated target object flow will be first repaired using the LAFC, and then the content is distributed among all of the video frames with restored object flows. Additionally, they utilize Flow-Guided Transformers to reconstruct the residual corrupted regions (FGT) or PEG.

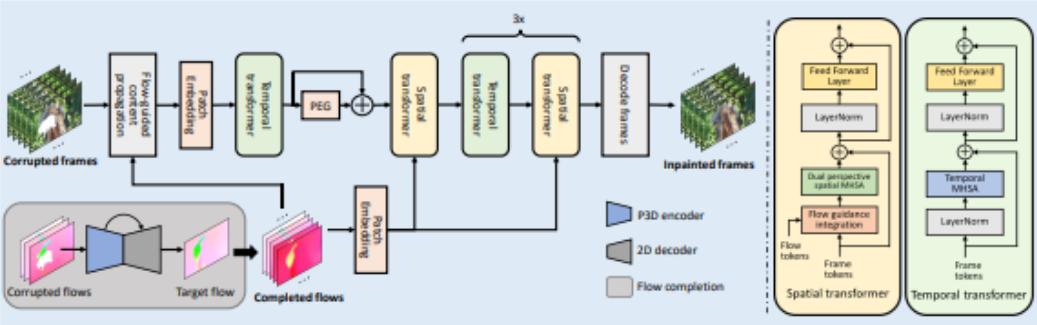


Figure 20: Flow-Guided Architecture

**Hao Ouyang et al. [17]** proposed a novel framework that works on video inpainting by adopting internal learning. The model brings in the novelty of dealing with challenges which mainly are detecting the object by a single binary mask and video inpainting of 4K resolution videos.

The model has the following flow in which it works. Cross frame referencing where in the object selected to be inpainted is coloured blue and further to red before finally generating a binary mask. This is then passed through a neural network to overfit the data and to improvise the overfitting the further procedure is Translational Equivalent Convolution which improvises the pixels of the image helping with better resolution.

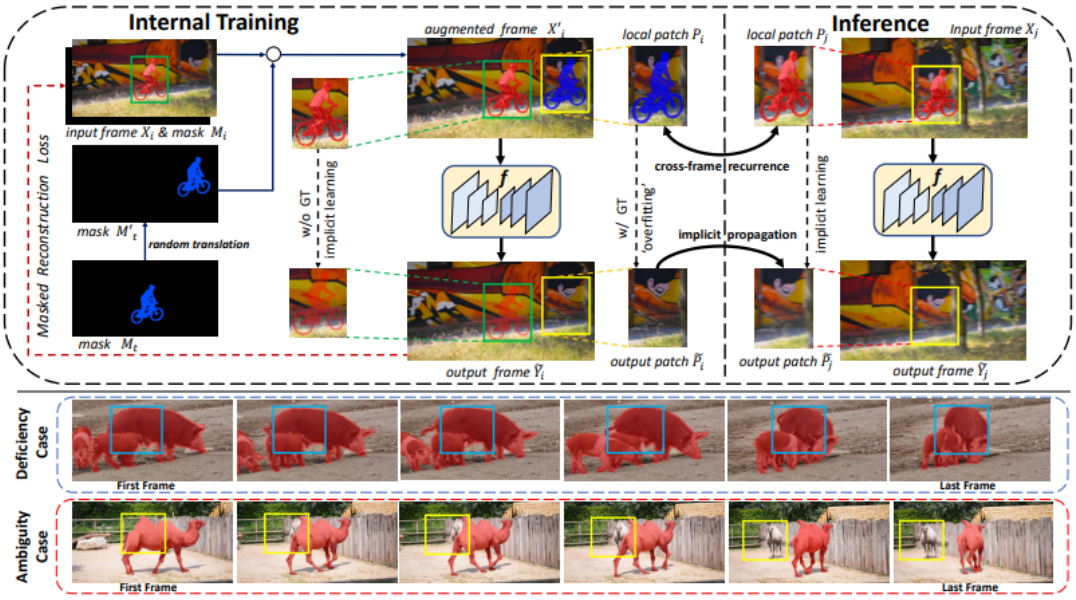


Figure 21 : Overview of the Proposed Model

## 3.4 Complete pipeline

A detailed research review on the novel pipeline designed to detect and inpaint the subtitle region in a video.

**Haoran Xu et al. [18]** proposed a novel pipeline to detect, remove and recognize subtitles in a video. The proposed system consists of 3 models, text detection, image inpainting and joined in a pipeline to accomplish the goal.

CTPN (Connectionist Text Proposal Network) model is used for text detection which is a fine tuned model on VGG-16. After gaining the coordinates of the bounding boxes around each subtitle line, the subtitle region is passed to CRNN for text recognition and simultaneously a contour of the subtitle region is generated which is passed to an

image inpainting network, EdgeConnect.

As a part of the post process, the recognized text is saved as a txt file and the inpainted image frames are joined back to video format with audio added back to it.

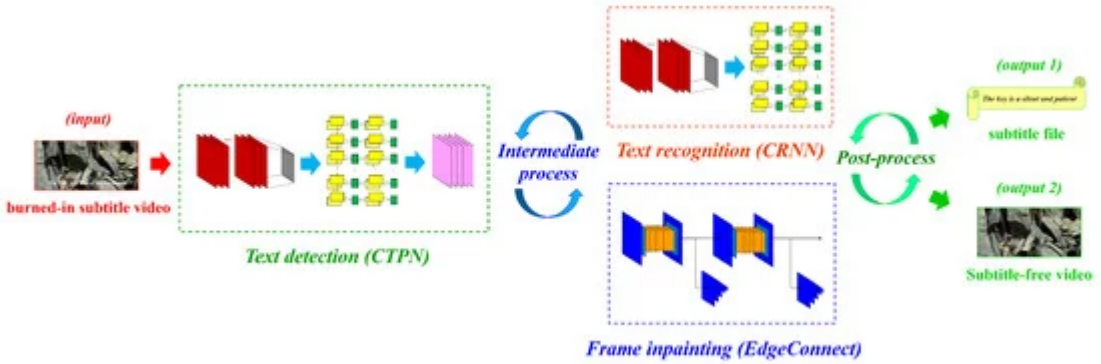


Figure 22: CTPN with EdgeConnect architecture

# CHAPTER 4

# DATA

This chapter explains the data generated for building the model. There are different versions and modifications to the dataset that have been introduced as time progressed, to train the model and also for effective testing purposes.

## 4.1 Overview

The dataset for this research is generated from multiple sources across the web. The dominant sources being YouTube, Prime Videos, and Netflix. Videos from different domains, such as movies, songs, tv shows, nature and classroom lectures, have been chosen to be a part of the dataset.

## 4.2 Datasets

The dataset includes videos consisting of subtitles in English, Hindi, Telugu and Malayalam [Devanagiri and Dravidian scripts]. For subtitle text detection, a more specific dataset is created. This dataset consists of image frames with scene text and high contrast background noise along with the subtitle text.

### 4.2.1 Video Dataset

The video dataset is a collection of video clips from multiple sources across the web. The videos have hard coded or embedded subtitles in the 4 languages and a subset of the same videos without the subtitles. Around 40 videos are generated in each language.

The dataset used is a self generated dataset that consists of 200 videos.

Each video clip has an average of 10 seconds video length. There are around an average of 300 frames in each video clip and the frame rate varies from 24 fps to 30 fps. Not all the frames in a video have the subtitles, which can be observed in movies and most of the videos.

The subtitles are white in colour and they are embedded into the video. The videos so generated have the subtitle only in the bottom end of the video and are in a horizontal fashion. They have a dynamic background i.e. they do not have the black mask behind it. The video has a complete subtitle in a single set of frames and it does not go word by word text. These are all the characteristics of the dataset generated.

### 4.2.1 Text Detection Dataset

The dataset generated for text detection is essentially a collection of image frames with complex backgrounds and embedded subtitle text. For this, a set of 50 videos with high contrast background noise and scene text, sometimes similar to subtitle text like news or twitter texts, are collected for each language.

These videos are then split into image frames and preprocessed. The segmented binary images, leaving out the frames without subtitles, are considered to be a part of the dataset. The ground truth of each of these image frames is the coordinates of the bounding box drawn around each subtitle text line in the image.

## 4.3 Statistics

| | |
|---|---|
| Subtitles | Embedded + Overlayed |
| Subtitle color | white (with/without black border) |
| Languages | English, Hindi, Telugu, Malayalam |
| Resolutions | 720p, 1080p |

Table 05: Dataset General Specifications

| | |
|---|---|
| Number of Videos | 200 |
| Number of Videos per language | 40 |
| Number of Videos for ground truth | 40 |
| Average Video Length | 10s |
| Average number of frames per video | 300 |
| Frame rate | 24 to 30 |

Table 06: Video Dataset Specifications

| | |
|---|---|
| Total number of image frames | 6336 |
| Total number of frames with subtitles | 5453 |
| Average number of frames per language | 1300 |

Table 07: Text Detection Dataset Specification

# CHAPTER 5

# METHODOLOGY

This chapter explains about the methodology used for detecting and removing the subtitle text from videos. The methodology involves a pipeline of 2 important models, a text detection model to detect the subtitle region in the video and a video inpainting model to fill the void created by subtitle text.

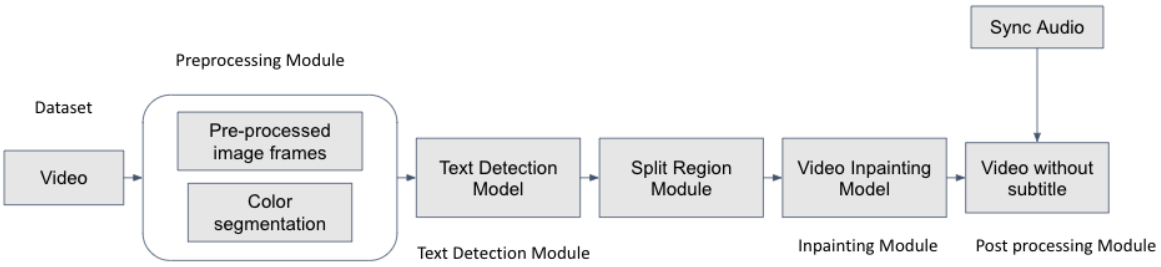


Figure 23: High level view of the pipeline

## 5.1 Pre - Processing

### 5.1.1 Splitting Video to Video Frames

The intake video is split into frames using the OpenCV library as the whole subtitle removal process becomes easier by being able to distinguish frames with/without subtitles and able to recognize the regions having subtitles. The metadata such as the filename, desired output format, dimension of the video, and frame rate of the video is recorded, so that the output video is in the desired format.

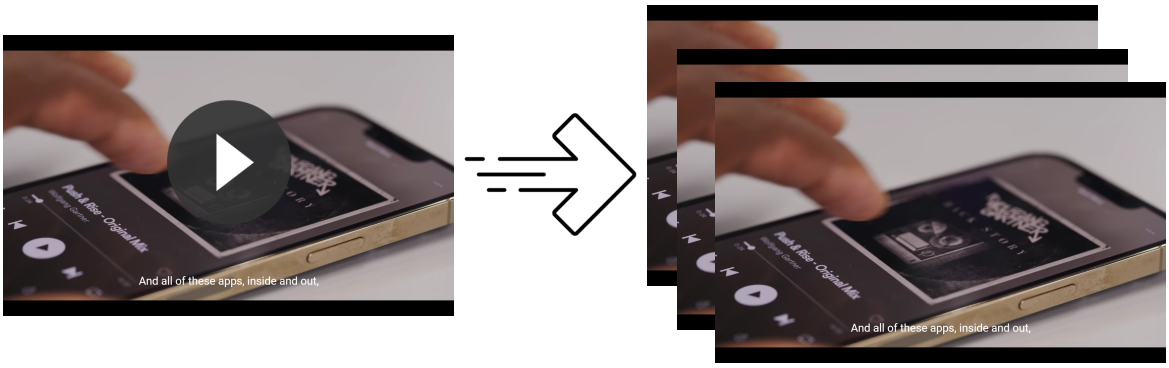


Figure 24: Splitting video into video frames

### 5.1.2 Image Segmentation

The frames obtained colour coding is changed from BGR (Blue Green Red) colour space to HSV (Hue Saturation Value) colour space, and binarized, regions with having HSV value within the limit ([0, 0, 200] and [150, 15, 255]) are set to [0, 0, 100] other regions are set to black [0, 0, 0].

The segmentation process overall improves the subtitle detection process as it removes the majority of scene text (non-subtitle regions) from the frame which will prevent it from inpainting other regions.

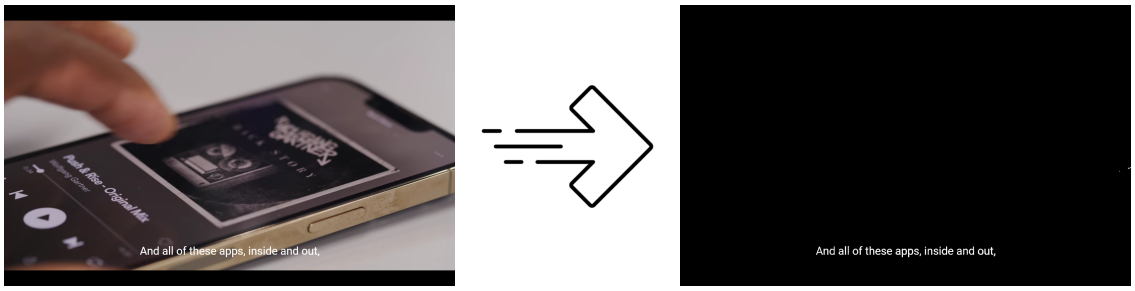


Figure 25: Frame Segmentation

## 5.2 Text Detection Module

A modified version of CTPN (Connectionist Text Proposal Network) model is used to detect the subtitle text over a complex background. This section entails the details about the dataset, architecture and the implementation of the text detection model.

CTPN is a novel text detection model that localizes scene text accurately and with good efficiency compared to other state of art models. It detects text in an image frame in 0.14s. However, this model is trained for localizing any kind of text in the image which is not desired as a property for subtitle text detection in the video since the background text (navigation boards, writing on blackboard or paper) is essential information that should not be eliminated from the video content. Therefore, we modified the model accordingly to detect subtitle text only.

### 5.2.1 Dataset

The dataset used for text detection is a collection of image frames with subtitles clipped from multiple videos. These videos are generated from various sources with a mix of subtitles in different languages, fonts and with/without background noise. Around 60 videos are generated in each language with an average length of 3s and frame rate of 30 fps. These videos are then converted to image frames and preprocessed. The final segmented binary format images are considered for training the text detection model. The rectangular boundary coordinates around the subtitle region serve as the ground truth.

### 5.2.2 Architecture

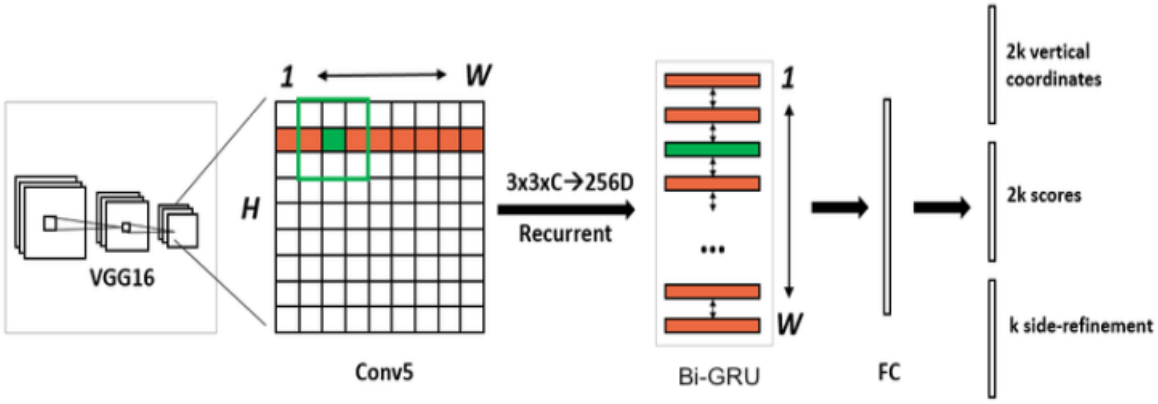


Figure 26: CTPN Architecture with Bi-GRU of Text Detection Module

CTPN uses VGG-16 deep network, which is an object detection model, to extract features from the image. Then, a 3 x 3 sliding window is slid through the W x H x C convolutional feature map yielding 3 x 3 x C features for prediction.

Since words are a series of characters, it is preferable to use a recurrent network to finely encode the sequential information of characters. The recurrent network consists of a 128D bidirectional GRU instead of bidirectional LSTM. This reduces the computational complexity of the model, as the GRU layer considers 2 gates as opposed to the 3 gates in LSTM.

Further down, the features are passed to a 512 dimension fully-connected layer similar to the CTPN model that is followed by a classifier and a regressor to predict the text proposal's scores (text or non-text) and the y-axis coordinates (ymin, ymax), and a refinement module to take faded text line edges into consideration.

*5.2.2.1 Code*

```
Base Model = models.vgg16(pretrained = False)

Layers = layers(Base Model) except the last one

RPN = Convolutional Layer (input = 512D, output = 512D, sliding window = 3x3)

RNN = GRU(input = 512D, 128, bidirectional=True)

FC = Convolutional Layer (input = 256D, output = 512D)

RPN classifier = Convolutional Layer (input = 512D, 10 * 2)

RPN regressor = Convolutional Layer(input = 512D, 10 * 2)
```

### 5.2.3 Training

The input to the model for training are segmented images which have subtitles and background noise as well. The corresponding labels are the bounding boxes in ICDAR dataset format around each subtitle text line in the image.

There are a total of 3.5k images for training across 4 languages. The image frames consist of subtitles in different regions of the frame, could be at top, bottom, left and right. The model is finetuned on our dataset for 10 epochs. Each epoch takes in a batch size of 1, therefore the total number of iterations being 46k iterations. The optimizer used for this training is SGD with OHEM algorithm.

### 5.2.4 Output and Split region

The final output of the CTPN model is the bounding coordinates around each subtitle text line. These coordinates (top left, top right, bottom left, bottom right) are used

to find the (xmin, ymin, xmax, ymax) bounds across all the frames which denote the common subtitle region across frames.

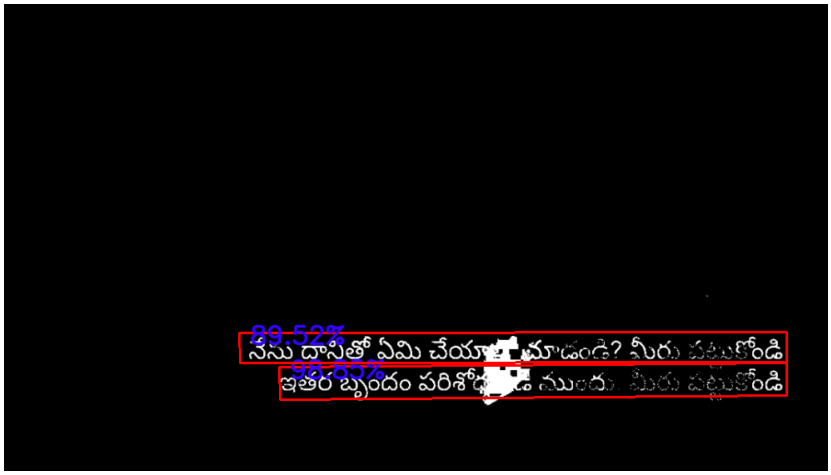


Figure 27: Detecting text in the segmented binary image using CTPN model

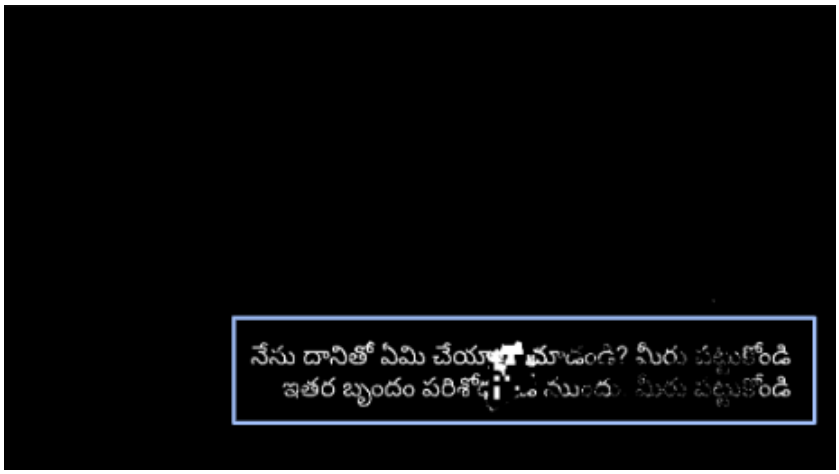

Figure 28: Generating the complete bounding box

The final bounds are taken into consideration to split the subtitle region into 240 x 432 dimension boxes. This is a novel way to inpaint each sub region taking advantage of the optical flow guided technique. Therefore, the output of this module is the set of sub region images and corresponding masks for each frame.


Figure 29: Splitting the subtitle region to 240 x 432 dimensions

## 5.3 Video Inpainting

End to End Framework of Flow Guided Video Inpainting is used to inpaint the region detected as subtitle from the text detection model. This section entails the details about the dataset, architecture and the implementation of the video inpainting model.

### 5.3.1 Dataset

The data used for video inpainting is the video sequence which is the input video and the binary masks generated in sequence (which is the output of the modified CTPN model)

### 5.3.2 Architecture

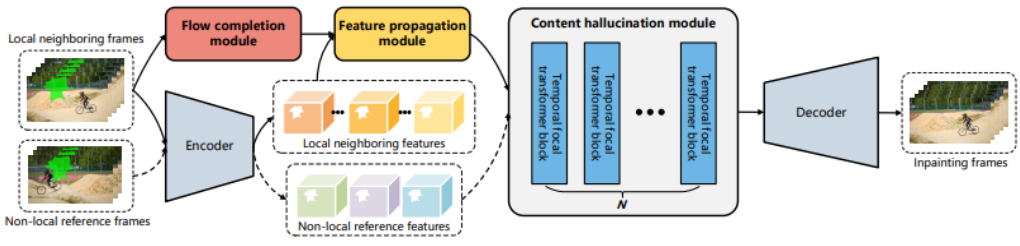


Figure 30: E2FGVI Architecture

The model has the following components.

They have mostly employed a context encoder, whose primary function is to transform erroneous frames into subpar frames with more cognitive efficiency for post processing. Through the flow completion module, identify and monitor the optical flow between immediate neighbors in further precision. Third, the entire optical flow aids feature alignment and bilateral dissemination and assists the feature extracted from immediate neighbors. Fourth, propagated local adjacent features and non-local reference features are coupled by cross-temporal focal transformers to generate content hallucination. The filled elements are then upscaled and reassembled into a final video scene by a decoder.

*5.3.2.1 Code*

```
Base_Model= E2EFVI(pretrained= True)

ref_frames=
get_reference_frames(frame_name,reference_no_of_frames)

mask=read_masks_from_folder(path_of_masks)

img_frame=frames_from_video(video)

mask_and_frame=gen_frame_and_mask(image,mask)

inpainted_frame=Base_Model.inpaint(image,mask)
```

5.3.3 Output

Figure 31: Inpainted Image Output

## 5.4 Construction of video from image frames

### 5.4.1 Rejoin the frames

After getting the inpainted frames from the inpainting model, we get those frames and in sequential order align them and rejoin it to the video based on its frame rate which is already obtained during the preprocessing stage. We use the python OpenCV function for the same.

### 5.4.2 Add audio

After getting the video from the previous step, we use the built in MoviePy library of Python and add audio to the video and then it would be specified in the output folder that is mentioned prior to the execution.


Figure 32: Inpainted video

# CHAPTER 6

# RESULTS AND DISCUSSION

In the entire dataset, 80% is training image frames, 10% is validation image frames and 10% is test image frames We calculate these accuracies using the Intersection over Union Method.

## 6.1 Text Detection

### 6.1.1 Intersection over Union [IoU]

The model efficiency is assessed by how well the model tests after training using **Intersection over Union (IoU).** The range of overlap between the predicted and ground truth coordinates is denoted by a value between zero and one.

$$IoU = \frac{Area\ of\ Intersection\ between\ two\ text\ bounding\ boxes}{Area\ of\ Union\ between\ two\ text\ bounding\ boxes} \times 100$$

### 6.1.2 Precision

Precision is a good measure to determine how much percentage of the region has been correctly deemed as the subtitle region in the whole output region of the text detection module.

$$Precision = \frac{True\ positive}{True\ positive + False\ positive}$$

### 6.1.3 Recall

Recall can find out what fraction of the ground truth is correctly detected as subtitles in the output region of the text detection module.

$$Recall = \frac{True\ positive}{True\ positive + False\ negative}$$

### 6.1.4 F1-Score

F1-score gives us a better representation on how good the model is by considering the values of both precision and recall.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

For Text Detection, we train our model and store the checkpoint of each epoch and then find out the train accuracy. The one that had the optimum train accuracy was used for the further working of the project.

| Epoch No | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| iou | 88.53 | 87.32 | 89.12 | 89.09 | 91.36 | 88.62 | 89.08 | 89.60 | 90.14 | 91.71 |
| precision | 98.06 | 89.80 | 91.90 | 91.34 | 93.14 | 90.11 | 91.69 | 90.84 | 91.54 | 92.94 |
| recall | 90.08 | 96.96 | 96.80 | 97.36 | 97.97 | 98.18 | 96.84 | 98.50 | 98.44 | 98.58 |
| f1-score | 93.34 | 93.02 | 94.14 | 94.04 | 95.34 | 93.62 | 94.03 | 94.08 | 94.19 | 95.46 |

Table 08: Train accuracies for different epochs

This above values are plotted on a graph to find the accuracy of the model and to detect the loss of the model

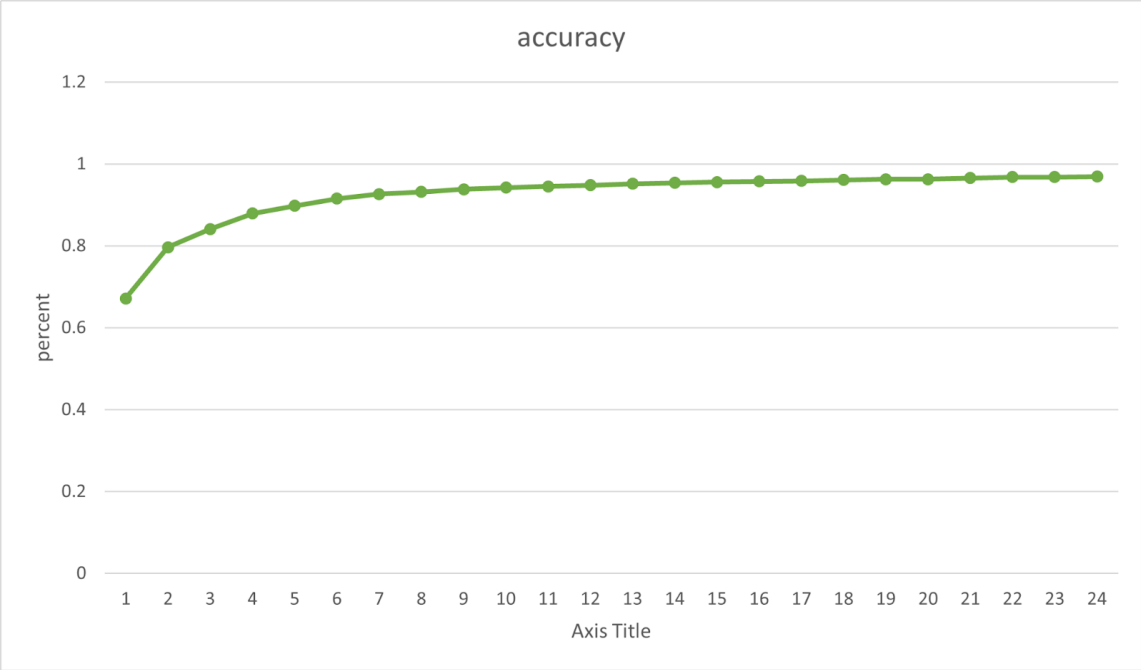


Figure 33: Model Accuracy


Figure 34: Model Loss

Based on the comparisons from the graph the Ideal epoch to consider for text detection was Epoch 09 which has an iou of 90.1943808413615 precision of 91.5242058800112 recall of 98.44121881722518 and an f1-score of 94.19297643343403 and based on this the test and validation metrics were calculated

| Type of data | iou | precision | recall | f1-score |
|---|---|---|---|---|
| Validation Data | 89.77 | 91.21 | 98.30 | 93.82 |
| Test Data | 90.12 | 91.88 | 98.10 | 94.37 |

Table 09: Text Detection Model Accuracy

## 6.2 Video Inpainting

### 6.2.1 Peak Signal to Noise Ratio [PSNR]

**Peak Signal to Noise Ratio (PSNR)** is the ratio of an image's maximum possible power to the power of corrupting noise, which influences the nature of its representation. To calculate the PSNR of an image, it must be compared to a desirable clean image with the largest achievable power.

### 6.2.2 Structural Similarity Index Measure [SSIM]

A technique for forecasting the perceived quality of digital television, cinematic visuals, and other types of digital images and videos is called the **Structural Similarity Index Measure (SSIM).** SSIM is a tool for calculating how similar two image frames are

to each other.

Here are the results obtained for inpainting. These values were obtained by comparing image frames of the original video and the inpainted video

| Testing Type | Testing Value |
|---|---|
| PSNR | 44.41697732249454 |
| SSIM | 0.9175597949261632 |

Table 10: Video Inpainting Model Accuracy

# CHAPTER 7

# CONCLUSION AND FUTURE WORK

The purpose of this problem statement is to detect multilingual subtitle text and remove them using video inpainting to obtain a subtitle free video.

- To detect only the subtitle text, our CTPN model is trained on binary masks of our in-house dataset.
- It works in high efficiency and precisely with an accuracy of 94%.
- The now obtained coordinates of the subtitle region is split into multiple inputs to maintain the resolution and fed as an input to an end to end network of flow guided video inpainting models.
- This model considers information from neighbouring frames in the video and uses them to successfully inpaint the subtitle text region.

# REFERENCES

[1] Zhi Tian , Weilin Huang , Tong He , Pan He , and Yu Qiao, *"Detecting Text in Natural Image with Connectionist Text Proposal Network"*, European Conference on Computer Vision,2016.

[2] Akhtar Jamil*, Jawad Rasheed, Bulent Bayram. "Local statistical features for multilingual artificial text detection from video images".* 2nd International Conference on Advanced Technologies, Computer Engineering and Science, Alanya, Turkey,2019

[3] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang. *"TextBoxes: A Fast Text Detector with a Single Deep Neural Network"*. Proceedings of the AAAI Conference on Artificial Intelligence 31,2016.

[4] Zhi Tian , Weilin Huang , Tong He , Pan He , and Yu Qiao, *"Detecting Text in Natural Image with Connectionist Text Proposal Network"*, 10th Spring Researchers Colloquium on Databases and Information Systems, Syrcodis 2014 Veliky Novgorod,2014.

[5] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, Hwalsuk Lee. *"Character Region Awareness for Text Detection"* . Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages. 9365-9374,2019.

[6] Shaswata Saha, Neelotpal Chakrabortya Soumyadeep Kundu, Sayantan Paul, Ayatullah Faruk Mollah, Subhadip Basu, Ram Sarkar, *"Multi-lingual scene text detection and language identification"*, (138), 2020

[7] Mohammad Khodadadi, Alireza Behrad, *"Text Localization, Extraction*

*and Inpainting in Color Images",* IEEE, 2012.

[8] Zhengmi Tang, Tomo Miyazaki, Yoshihiro Sugaya, Shinichiro Omachi, *"Stroke-Based Scene Text Erasing Using Synthetic Data for Training"*, 2021

[9] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. *"Resolution-robust large mask inpainting with fourier convolutions"*. arXiv preprint arXiv:2109.07161, 2021.

[10] Kamyar Nazeri, Eric Ng, Tony Joseph and Faisal Z. Qureshi. "*Edgeconnect: Generative image inpainting with adversarial edge learning"*, 2019.

[11] Guilin Liu, Fitsum A. Reda, Kevin Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. "*Image inpainting for irregular holes using partial convolutions"*. In The European Conference on Computer Vision (ECCV), volume 11215, pages 89–105, 2018.

[12] Dr. A. Pasumpon Pandian, *"Image Inpainting Technique for High quality and Resolution enhanced Image creation"*, Journal of Innovative Image Processing, 2019

[13] Zhen Li, Cheng-Ze Lu, Jianhua Qin, Chun-Le Guo, Ming-Ming Cheng. *"Towards An End-to-End Framework for Flow-Guided Video Inpainting"*. CVPR 2022,2022.

[14] Sungho Lee, Seoung Wug Oh, DaeYeun Won, Seon Joo Kim. *"Copy-and-Paste Networks for Deep Video Inpainting"*.ICCV,2019.

[15] Yanhong Zeng, Jianlong Fu, Hongyang Chao.*"Learning Joint Spatial-Temporal Transformations for Video Inpainting"*.ECCV2020,2020.

[16] Kaidong Zhang , Jingjing Fu , and Dong Liu. *"Flow-Guided Transformer for Video Inpainting"*. arXiv preprint arXiv:2208.06768, 2022.

[17] Hao Ouyang, Tengfei Wang, Qifeng Chen. *"Internal Video Inpainting by Implicit Long-range Propagation"* . Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) pages. 14579-14588, 2021

[18] Haoran Xu , Yanbai He , Xinya Li , Xiaoying Hu , Chuanyan Hao, Bo Jiang , "*Joint Subtitle Extraction and Frame In-painting for Videos with Burned-In Subtitles*", Special Issue Recent Advances in Video Compression and Coding, 2021.

[19] Prof. Abhishek Mehta, Dr. Ashish Chaturvedi, "Extraction and Recognition of Handwritten Hindi and Gujarati Character Using Artificial Neural network Approach", JETIR, (6),2019

[20 ]Omar Elharroussa, Noor Almaadeeda, Somaya Al-Maadeeda, Younes Akbaria, "*Image inpainting: A review*", 2019

# APPENDIX: DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

ICDAR- International Conference on Document Analysis and Recognition

E2FGVI - End To End Framework for Flow-Guided Video Inpainting

GAN - Generative Adversarial Network

CRNN - Convolutional Recurrent Neural Network

CTPN - Connectionist Text Proposal Network

LSTM - Long Short-Term Memory

IoU - Intersection over Union

MSER - Maximally Stable Extremal Region

SWT - Stroke Width Transform

CNN - Convolutional Neural Network

SSFT - Superpixel based Stroke Feature Transform

DLRC - Deep Learning based Region Classification

CTR - Candidate Text Regions

STBB - Subtitle Top Bottom Boundary

SCW - Single Character Width

SLRB - Subtitle Left Right Boundary

FPN - Feature Pyramid Network

DFC - Deep Flow Completion Network

HFEM - Hard Flow Example Mining

IEEE - Institute of Electrical and Electronics Engineers.

IJERT - International Journal of Engineering Research and Technology

JETIR - Journal of Emerging Technologies and Innovative Research

TPAMI - Transactions on Pattern Analysis and Machine Intelligence.

ICCV - International Conference on Computer Vision

ECCV - European Conference on Computer Vision

DIP - Deep Image Prior

IPCV - Image Processing and Computer Vision

## Subtitles

ORIGINALITY REPORT

| 7% | 4% | 5% | 1% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | www.arxiv-vanity.com<br>Internet Source | 1% |
| 2 | web.archive.org<br>Internet Source | 1% |
| 3 | Neethu M Sathyan, Sashi Rekha Karthikeyan. "Infrared Thermal Image Enhancement in Cold Spot Detection of Condenser Air Ingress", Traitement du Signal, 2022<br>Publication | 1% |
| 4 | Mohammad Khodadadi, Alireza Behrad. "Text localization, extraction and inpainting in color images", 20th Iranian Conference on Electrical Engineering (ICEE2012), 2012<br>Publication | 1% |
| 5 | "Advances in Visual Computing", Springer Science and Business Media LLC, 2019<br>Publication | <1% |
| 6 | Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, Hwalsuk Lee. "Character Region Awareness for Text Detection", 2019 IEEE/CVF Conference on Computer Vision and Pattern | <1% |

Recognition (CVPR), 2019
Publication

| | | |
|---|---|---|
| 7 | Submitted to University of Greenwich<br>Student Paper | <1% |
| 8 | Submitted to Queensland University of Technology<br>Student Paper | <1% |
| 9 | docs.cloudimage.io<br>Internet Source | <1% |
| 10 | docs.opencv.org<br>Internet Source | <1% |
| 11 | pyimagesearch.com<br>Internet Source | <1% |
| 12 | www.itu.int<br>Internet Source | <1% |
| 13 | cdn.iiit.ac.in<br>Internet Source | <1% |
| 14 | dspace.lboro.ac.uk<br>Internet Source | <1% |
| 15 | dspace.ut.ee<br>Internet Source | <1% |
| 16 | nova.newcastle.edu.au<br>Internet Source | <1% |
| 17 | refbase.cvc.uab.es<br>Internet Source | <1% |

| | | |
|---|---|---|
| 18 | "Computer Vision – ECCV 2016 Workshops", Springer Nature, 2016<br>Publication | <1 % |
| 19 | Zhengmi Tang, Tomo Miyazaki, Yoshihiro Sugaya, Shinichiro Omachi. "Stroke-Based Scene Text Erasing Using Synthetic Data for Training", IEEE Transactions on Image Processing, 2021<br>Publication | <1 % |

| | | | |
|---|---|---|---|
| Exclude quotes | On | Exclude matches | < 5 words |
| Exclude bibliography | On | | |