Section A-Research paper
ISSN 2063-5346

Multilingual Subtitle Detection and Removal using Video Inpainting

[1]Rishab Santosh, [2]Ratna Bojja, [3]Sahana B Manjunath, [4]Ramya C,[5]Badri Prasad

[1,2,3,4]Student, PES University, RR Campus, Bangalore, [5] Assistant Professor, PES University, RR Campus, Bangalore

Email: [1]rishab.santosh189@gmail.com, [2]ratna.bojja01@gmail.com,
[3]mounabhargava1601@gmail.com,[4]ramyashekar2401@gmail.com,[5]badriprasad@pes.edu

Contact: [1]9148766038, [2]8095631985,[3]7019555270,[4]8431048596,[5]9845914285

***Abstract*—A novel end-to-end framework is proposed for the detection and removal of hardcoded or embedded subtitles from videos. The framework utilizes two deep learning models, a modified CTPN (Connectionist Text Proposal Network) model, and an E2FGVI (End-to-end Framework for Flow Guided Video Inpainting) model, to achieve its goal. The modified CTPN model is designed to detect multi-scale, multilingual subtitle text while ignoring background or scene text and has been optimized for faster detection, reducing the computational cost of detecting subtitle text from 0.14s/frame to 0.08s/frame. The output video resolution is preserved, ensuring that the final output is of high quality. The proposed framework represents a valuable contribution to the field of video processing, as it addresses the challenges of detecting and removing subtitles while preserving the content of the original video. The entire process of detecting and removing subtitles can be completed in approximately 120 seconds, making it a computationally efficient solution.**

Keywords—hardcoded subtitles, scene text, inpainting, deep learning, multi-scale, multilingual, video processing.

1. INTRODUCTION

Subtitles are 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 a 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 the preferred language, on the video for better understanding.

The removal of the embedded subtitles from the video leads to the 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 the 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, the current approach 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 the necessary content. Our contributions include building an end-to-end framework for detecting multilingual subtitles in the video and removing them using the video inpainting technique. The various stages in the pipeline to achieve this goal are

specified as preprocessing, text detection, video inpainting, and postprocessing.

In the pre-processing stage, we convert the video into image frames and extract audio from it. They are named in serial order to avoid misinterpretation of the frames. Subtitles are high-intensity/contrast components in videos. Therefore the image frames are further transformed from BGR to HSI model. With thresholding, based on the intensity and saturation thresholds, the low-contrast components, like part of the scene text, are eliminated and the image is converted to a binary format (segmented image).

For detecting the subtitle text, excluding the scene text, a modified CTPN (Connectionist Text Proposal Network) model is employed, which has been trained on an in-house dataset of 3.5k binary images containing subtitle text, background noise, and scene text. The output of this model is the set of coordinates of the bounding box around the subtitle region, which is used to generate a subtitle mask.

In the video inpainting stage, the subtitle region and mask from the text detection stage are split into smaller segments and sequentially passed to an End to End Framework for Flow Guided Video Inpainting model for inpainting. This process is performed to preserve the resolution of the video and improve the speed of the inpainting process. Each smaller segment is inpainted and joined back to the original frame to get a subtitle-free frame.

Finally, in the post-processing stage, the inpainted frames are reconstructed into a video and synchronized with the audio to obtain the final output video without subtitles.

2. LITERATURE SURVEY

Removal of subtitles is a challenging task that is not addressed by many research papers. However, a lot of research work is done on the submodules such as thresholding, text detection model, and video inpainting model, that help in solving this problem.

A.Jamil et al. [2] explained the use of statistical features to detect multilingual artificial text in image frames with complex backgrounds. Subtitles being the high-contrast objects can be retrieved using segmentation techniques such as thresholding. These techniques retain other high-contrast residual elements such as scene text and background objects. The segmented image is essentially a binary mask with subtitles and other background noise. To filter this noise, a text detection model is used. Mikhail Zarechensky et al. [21] 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, the number of successive images detected in a sequence where this region stays the same. We can choose these regions called Maximally Stable Extremal Regions or MSER. Zhi Tian et al. [1] proposed a novel deep learning architecture (CTPN) to detect scene text. This model, modified to detect only subtitle text in the current study, ensures that multilingual and multiscale subtitle text is detected accurately.

Zhen Lil et al.[13] proposed a model wherein the model works on object flow across frames and inpainting methods through propagation pixels. The model used the video and binary masks of the video in sequential order so that the model inpaints accordingly and delivers the inpainted video as the output. Haoran Xu et al. [18] proposed a novel framework for detecting, removing, and recognizing subtitles in a video. The proposed system consists of three models, text detection (CTPN), image inpainting (EdgeConnect) and CRNN joined in a pipeline to accomplish the goal. This framework gave the backbone idea to construct the pipeline to detect and remove subtitles. The original pipeline, mentioned in the paper, has been modified to include video inpainting rather than image inpainting in the current study, as the experimental phase of the project explored the glitches in the output video due to the frame-by-frame image inpainting, which has been improved in video inpainting.

For Testing, Zhi Tian et al[1] uses Iou for testing the accuracy of the model.They basically used the area covered by both the regions of the ground truth and the output as the area of intersection and the collective area as the area of union for the calculation of intersection over the union. [1],[5],[6] uses Precision, recall and f1 score for testing.

The quality of Dr. A. Pasumpon Pandian's [12] proposed image in-painting technique is assessed by utilizing two popular metrics in image processing: the Structural Similarity Index (SSIM) and the Peak Signal-to-Noise Ratio (PSNR). The SSIM measures the likeness between the original and in-painted images, while the PSNR evaluates the difference in terms of noise or distortion. The outcome of these assessments provides an indication of the image quality of the in-painting technique, and its effectiveness and efficiency. These metrics aided us in effectively assessing our methods and making enhancements to them.

3. DATASET

The dataset for this research is generated from multiple sources across the web. The dominant sources are 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.

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.

a. 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 of video length. There are 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 subtitles, which can be observed in movies and most videos.

The subtitles are white in color and they are embedded into the video. The videos so generated have the subtitle only at 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.

b. Text Detection Dataset

To train and evaluate the model, a dataset of image frames with clipped subtitles from a diverse range of videos was created. This dataset was composed of videos in various languages, each with a distinct font style and varying levels of background noise. Approximately 60 videos were generated for each language, with an average length of 3 seconds and a frame rate of 30 fps. The videos were then transformed into image frames and preprocessed to obtain a final binary format suitable for training. The model was trained using these image frames, with the rectangular boundary coordinates around the subtitle region serving as the ground truth label.

c. Statistics

Subtitles	Embedded + Overlayed
Subtitle color	white (with/without black

	border)
Languages	English, Hindi, Telugu, Malayalam
Resolutions	720p, 1080p

TABLE I: 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	24-30

TABLE II: Video Dataset Specifications

4. METHODOLOGY

a. Pre - Processing

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.

The frames obtained color coding is changed from BGR (Blue Green Red) color space to HSV (Hue Saturation Value) color 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.

b. Text Detection

The Connectionist Text Proposal Network (CTPN) model has been adapted for our use case of detecting subtitle text in videos. The model leverages its text localization accuracy and efficiency, being able to detect text in an image frame within 0.14 seconds. However, the original model was trained to detect any type of text in an image, including essential background information such as navigation boards and writing on blackboards or paper. To ensure that the video content is not altered, our modified CTPN model has been specifically designed to detect only subtitles in the video, while disregarding other types of text.

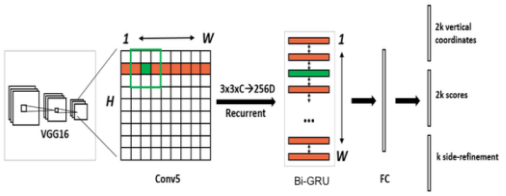

Fig I: CTPN Architecture

Our modified version of the CTPN model utilizes a VGG-16 deep network to extract features from each image frame in the video. The features are then processed through a 3x3 sliding window to identify potential subtitle regions. In order to accurately capture the sequential information of characters within words, we have employed a 128-dimensional bidirectional GRU instead of a bidirectional LSTM as the recurrent network. This change reduces the computational complexity while still providing effective character encoding. The output features are then passed to a 512-dimensional fully-connected layer and evaluated by a classifier and regressor to predict the scores for text vs. non-text and the y-axis coordinates (ymin and ymax) of the subtitles. Additionally, we have incorporated a refinement module to consider faded text lines and enhance the accuracy of text detection.

The CTPN model was trained on a dataset consisting of 3.5k segmented images collected from various videos in 4 different languages. Each image contained both the subtitle text and background noise. The labels for the training data were the bounding boxes around each subtitle text line in the ICDAR format. To fine-tune the model on our dataset, the model underwent 10 epochs with a batch size of 1, resulting in 46k iterations. The optimization was performed using the SGD algorithm combined with the Online Hard Example Mining (OHEM) technique. The output of the CTPN model is the bounding coordinates around each subtitle text line, which were then used to determine the common subtitle region across all the frames. The common region was further divided into 240 x 432 dimension sub-regions, which were then inpainted using an optical flow-guided technique. This resulted in a set of sub-region images and corresponding masks for each frame.

Fig II: Splitting the subtitle region to 240 x 432 dimensions

c. Video Inpainting and Post-Processing

We have used the End to End Framework of Flow-Guided Video Inpainting to inpaint the video to remove the subtitles with the binary mask that is obtained from the Text Detection Model.

The dataset used for this model is the video that was collected from the in-house dataset. For that video, the respective binary masks in the sequential model are obtained from the Text Detection Model. This is passed to the E2EFGVI for inpainting.

The End to End Framework of Flow-Guided Video Inpainting has a frame-level content encoder which is used for encoding the frames that have subtitles (maligned frames). This is achieved by converting those frames into a lesser resolution frame which aids in reducing the cost of computation of the model and also in further processing of the model. Then a flow completion model is used to restore the temporal information and also capture the optical flow from the nearby frames and similar further frames. Then is a feature propagation model which helps in getting the prominent and primary features from the frames referenced by the flow completion component. This is to achieve feature alignment and bi-directional propagation. Further, there is a content hallucination model which primarily is multi-layer temporal focus transformers that perform content hallucination by combining the obtained propagated local neighboring frames with non-local reference features. Finally, the architecture concludes with a decoder which is used to scale up the inpainted features and reconstruct it back to the video.

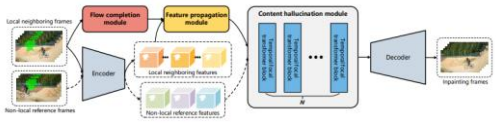

Fig III: E2FGVI Architecture

We pass the binary mask and the video as input to the Inpainting model. The model first converts the video into image frames and splits the subtitle region into dimensions of 240 x 432. For accurate inpainting, we split the binary masks as well. All these split regions with their respective binary masks are passed to the E2EFGVI model where the split frames are inpainted.

d. Post-Processing

In the Post-Processing module, we join the split sub-regions and combine them to form the inpainted subtitle region of the image. Then this part of the strip is overridden on the original image frame hence not entirely compromising on the image resolution. These image frames are arranged in sequence and then they are converted to a video. Then we sync the audio that is obtained during the pre-processing module, via the Python library MoviePy. The final video will be stored in the output path given by the user initially.

.

RESULTS AND DISCUSSION

We had an in-house dataset that consisted of two hundred videos. Each language had forty videos and the rest forty were the ground truth videos. Each of them had a resolution of either 720p or 1080p. The average duration of each of these videos was 11 seconds with a frame rate of thirty frames per second. Other than this, to train our text detection model, we used 3.5K images of the four languages and divided them into train, validation, and test data in the ratio 80:10:10. To generate these frames, we generated videos and then converted them into frames because in a video there would be many duplicate frames which would further help with the learning of the model.

These images that were passed through the text detection model, were segmented to achieve better accuracy and confidence level. The dataset was made more challenging by the addition of subtitles at various regions (top-left, top, top-right, bottom-right, bottom, bottom-left). Some edge cases were also considered wherein the watermark or the scene text was also retained. We passed these images through the text detection model and then compared the results with the ground truth we had collected previously.

For the inpainting, we used the videos that were generated before and compared them with the ground truth to calculate the accuracy of the pipeline.

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 III: Train accuracies for different epochs

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

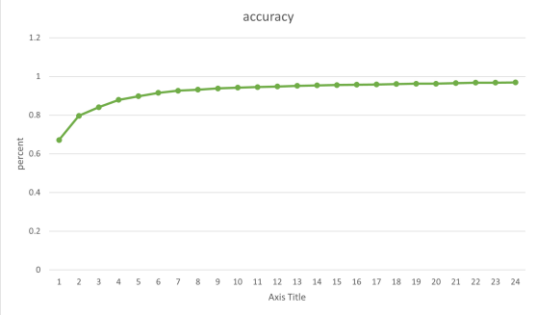

Fig IV : Model Accuracy

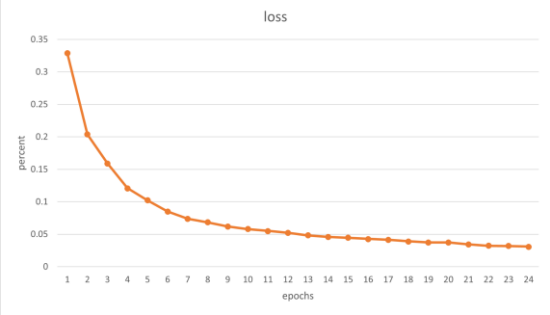

Fig V: Model Loss

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 IV: Text Detection Model Accuracy

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 V: Video Inpainting Model Accuracy

We tested our model on certain edge cases such as news videos, and lecture videos, and the model's functioning was accurate and it didn't fail those edge cases.

We also tested the results with other horizontally written languages such as Urdu, Bengali, Russian, and French and the model worked fairly well with them as well.

LIMITATIONS

Though the model works with greater accuracy, there are certain restrictions and constraints to the model that we haven't worked on or trained the same to improvise it better.

The first one being our model was trained and tested on videos that only had white-colored subtitles. Hence its performance is ambiguous when the video has different colored subtitles. Further, we tested our model with videos that had subtitles written horizontally only. Hence the model's function on subtitles that are present vertically is still something that the model has to learn and hence something that we have to work on.

CONCLUSIONS AND FUTURE WORK

The purpose of the problem statement is to detect multilingual subtitle text and removal using video inpainting to obtain a subtitle-free video. The CTPN model is trained on binary masks of our in-house dataset and has an accuracy of 94%. The obtained coordinates of the subtitle region are split into multiple inputs to maintain the resolution and fed as an end-to-end network of flow-guided video inpainting model. The text detection model can be made to work on upstanding subtitles, different fonts, and color subtitles even vertically subtitled videos can be considered.

The video inpainting module can be improved by considering side-by-side subtitle sub-regions to be spatially consistent. Further, our inpainting model can be improvised by considering KNN algorithm for inpainting. Instead of using the sliding window and collecting redundant frames or leaving out any useful information away and considering frames after a certain serial number, we can use KNN to calculate pixel densities and collect the ones that match the

frame's pixel density.

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 31,216.

[4] 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.

[5] 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.

[6] Mohammad Khodadadi, Alireza Behrad, *"Text Localization, Extraction and Inpainting in Color Images",* IEEE, 2012.

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

[8] 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.

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

[10] 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.

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

[12] 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.

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

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

[15] Kaidong Zhang, Jingjing Fu, and Dong Liu. *"Flow-Guided Transformer for Video Inpainting"*, Artificial Neural network Approach", JETIR, (6),2019

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

[17] Mikhail Zarechensky, Vassilieva N, *"Text Detection in Natural Scenes with Multilingual Text", 10th Spring Researchers Colloquium on Databases and information systems*, Syrcodis pages 32-35, (10), May 2014.