TEXT DETECTION AND EXTRACTION FROM VIDEO USING DEEPLEARNING TECHNIQUES

A THESIS SUBMITTED TO
DEFENCE INSTITUTE OF ADVANCED TECHNOLOGY , PUNE
FOR THE SEMESTER THREE EVALUATION (2021-2023) OF
MASTER OF TECHNOLOGY
IN
DATA SCIENCE

BY
RAMKRISHANA MARRI
(Roll No. 21-27-21)

UNDER THE SUPERVISION OF
Dr.Odelu Ojjela

SCHOOL OF COMPUTER ENGINEERING
&
MATHEMATICAL SCIENCES
DEFENCE INSTITUTE OF ADVANCED TECHNOLOGY
(DIAT),
PUNE, INDIA

DEC 2022

Dedicated to

I would like to dedicate this thesis work to my family members, Teachers and each and everyone who supported and motivated me

C E R T I F I C A T E

This is to certify that the Thesis entitled "**Text Detection and Extraction from Video using Deep Learning Techniques** " submitted to Defence Institute of Advanced Technology, Girinagar, for the award of the degree of ***Master of Technology***, is the bonafide research work done by **Mr. RAMKRISHANA MARRI** under my supervision. The contents of this thesis have not been submitted elsewhere for the award of any degree.

Dr Odelu Ojjela
Asso.Professor
Defence Institute of Advanced Technology
Girinagar, INDIA

.

DECLARATION

This is to certify that the work presented in the Thesis entitled **"Text Detection and Extraction from Video using Deep Learning Techniques"**, is a bonafide work done by me under the supervision of **Dr Odelu Ojjela** and has not been submitted elsewhere for the award of any degree.

Date:_____

Place:_____

Name : RAMKRISHANA MARRI

Roll No. : 21-27-21

Department : So CE & MS

Defence Institute of Advanced Technology

Girinagar, Pune

COUNTERSIGNED

INTERNAL SUPERVISOR	EXTERNAL SUPERVISOR
Dr Odelu Ojjela	Dr J V SatyaNarayna
Asso.Professor	Scientist G
DIAT PUNE	RCI DRDO

ACKNOWLEDGEMENTS

I would like to convey my heartfelt gratitude to Dr. J V SatyaNarayana. His efforts to help me with my thesis have been the only thing that has kept me motivated and on task. He's continuously challenged me to think critically and pick up new information. At the same time, he has also been a calming influence when there have been setbacks in the research work. I would like to thank Dr.Odelu Ojjela for his tremendous support and guidance to define the problem statement and for mentoring me to learn and work on the project for M.Tech in Data Science.

I would also like to express my gratitude to my colleagues for their valuable discussion on the topics.

A B S T R A C T

With advanced technology, smart devices and high- speed internet Textual content appearing in semantic retrieval of videos, live stream videos, as well as YouTube videos. The current research offers a thorough framework for finding and identifying text in video frames. Videos have shown to be a fantastic source of knowledge. There is a tonne of information and data in the video's text, but it could not be changed. If this content could be edited, that would be helpful information for us. The primary task at hand right now is to extract the text from the video. Deep learning techniques are the foundation for text recognition from video frames. Convolutional neural networks (CNN) and long short-term memory (LSTM) networks were combined for deep learning. People should provide as input the video he/she wants to capture the text. The technology analyses the video and produces the text in a text file that may be edited.

Keywords— OCR, Frames, text detection, Text Recognition, LSTM, CNN.

Contents

Chapter 1

Introduction and Literature Review

1.1 Introduction

With advanced technology and increasing internet speed, interest of people shifting from television to YouTube. The biggest benefit of YouTube over TV is that it provides individuals with the necessary videos. People have a limited amount of time because TV is fixed according to their schedule. As the focus shifts to YouTube, the technology now being developed enables individuals to efficiently and quickly access the information present in the text of these films. The current approach transforms the video's content into editable form, which is then saved in a text file.Videos related to news, science, health, politics, and education are frequently found on YouTube. Videos have shown to be a fantastic source of knowledge. There is a tonne of information and data in the video's text, but it could not be changed. This text will be helpful information for us, it will be kept effectively, and it will be simpler to access it in the future if it is changed to editable form. After seeing an educational film once, a person could decide not to view it again because he has already seen it in its entirety and has read the key points, which have already covered the subject. At that point, the suggested solution facilitates information access for the public by transforming the video's text into an editable form. text files that are editable The key benefit of the text file is that it takes less time to read than a movie. Additionally, if users wish to add any additional information that is not possible to include in a video but is possible in a text file, they can edit the content in the text according to their needs.

The proposed system's operation is simple and user-friendly. if someone gets a video from a website, such as YouTube, from which they want to extract the text. The proposed system take video as input and divides the (video)movie into individual frames and performs text extraction and detection on each frame. Each frame's identified text is saved in a .txt file.

Highlights

- The given input is converted into video frames
- Different filters are applied on each frame before text extraction
- Tesseract-OCR was used to extract texts from each video frame

Requirements:

- OpenCV
- Python
- Tesseract-OCR

1.2 Literature Review

Baseem Bouaziz, Tarek Zlitni, Walid Mahdi [2] automated video text extraction explained It does video indexing based on content. This approach can only find static text that has been overlaid.

Datong Chen, Jean-Marc Odobez [3]have put forth a technique that reduces character mistake rates and eliminates character noise, both of which significantly impair optical character recognition.

Lifang Gu [5] text detection in MPEG (Moving Picture Experts Group) video frames was discussed. Redundancies in spatial and temporal data are decreased. Only MPEG videos can be converted using this method.

Anubhav Kumar, Neeta Awasthi [6], have suggested a technique to localise the text information in both image and video formats. It is simple to identify and extract text from photos, but more challenging to do so when a video is playing. It is simpler to extract the text files from the multimedia file once the machine finds them. This system's disadvantage is that processing lengthy videos takes a very long period.

Punit Kumar, P. S. Puttaswamy [7] have suggested a solution that applies area-based filtering to get rid of noisy blobs in the image. When the background exhibits sharper changes in intensity, this strategy is ineffective.

Chapter 2

Problem Statement And System Overview

2.1 Problem Statement

There are two types of text occurring in a video

- Natural text
- Superimposed text

2.1.1 Natural Text:

Natural text is the text that appears in the video while it is being recorded. These texts are part of a video recorded scene [2]

Example: Flat Number, Vehicle Number Plate.

Figure 2.1: Figure Natural Text

2.1.2 Superimposed text:

Superimposed text: Superimposed text is text that is added to video after it has been shot but before it was actually part of the scene. Ex: Text that appears in a news video. [2]

Figure 2.2:

2.2 System Overview

The proposed system has three main components:

- Frame Generation
- Text Recognition and Extraction
- Text File Generation

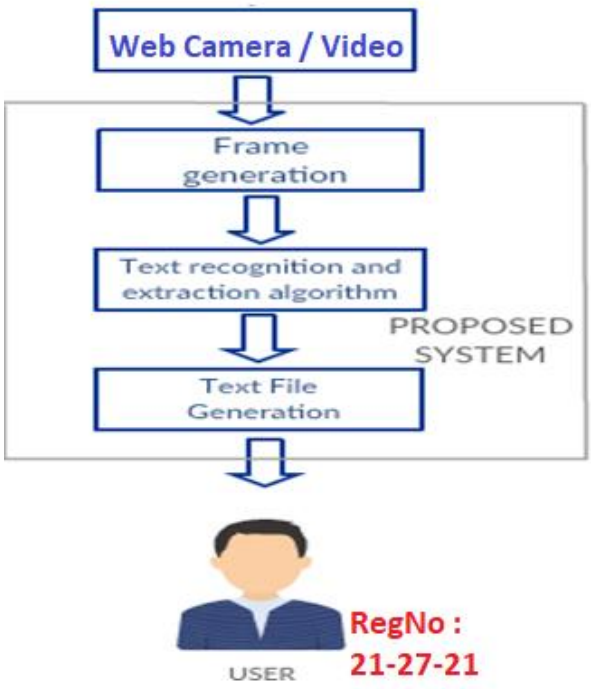

Figure 2.3: System Overview [1]

2.2.1 Frame Generation:

The video is turned into frames in this stage. Images from a certain point in a video are called frames. The frames are generated at regular intervals so that text in the following frame doesn't repeat itself a lot. Any picture format can be used to save these frames. When converting video to frames, the users will have two choices. The first is to convert the whole video, and the second is to convert a particular section of the video. People will

choose option one if they want text from the entire video. People will choose the second option, which allows them to choose the start time and finish time for text extraction, if they just require text from a specific time period. The chosen video clip will be converted into photos and kept in a separate folder for convenient access while a text extraction technique is applied to it.

2.2.2 Text Recognition and Extraction:

This process is carried out for each frame. Using the algorithm explained in the following section, the text region is found in this phase. To improve the effectiveness of text extraction, the discovered text sections are then refined. On the regions that have been identified, the text extraction method is applied. Font colour, text size, backdrop colour, and video resolution all have an impact on how well text is detected. [3].

2.2.3 Text File Generation:

A text file is used to hold the retrieved text. The created text is added to the preceding text in the text file and stored for each frame. People will be provided the path to the result file after all of the photos' text has been extracted. The text file is considerably smaller in size than the movie. As a result, information may be accessed more quickly and memory is conserved.

Chapter 3

Methodology

The suggested framework is described in detail in this section and is illustrated in Fig. 6. The text detector, script identifier, and text recognizer are the three key components that make up the entire system. A wide range of systems, such as content summarization, keyword-based alert generation, and indexing and retrieval, can be built on top of these modules at the application layer. The first module, text detector, is required to locate and recognise every piece of text in a frame. Since text can appear in many scripts (within the same frame), the text portions that have been found are provided to the script identification module, which splits the text lines according to the script (English and other Indian Language being the two scripts considered in the present study). The text is then forwarded to the respective recognition engines of each script, where they convert the images of the text lines into strings that can be used to a range of situations.

3.1 workflow

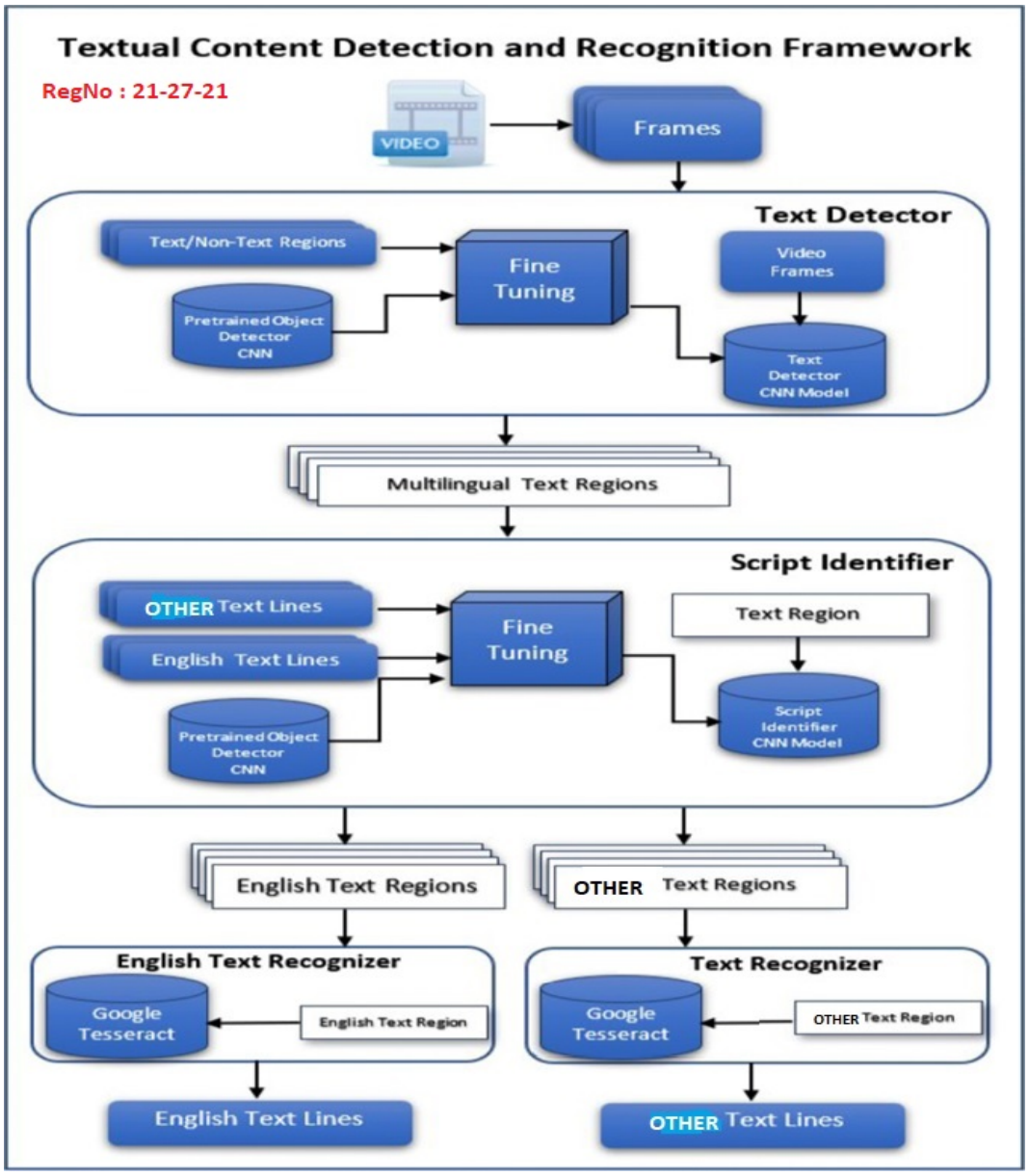

Figure 3.1: workflow [9]

3.2 Text detection :

The proposed framework's first step is the identification of potential text sections in the retrieved video frames. Modern object detectors based on convolutional neural networks

(CNN) have been used to recognise textual information in a given frame. Despite the fact that many object detectors are highly accurate in detecting and recognising various objects and have been trained on tens of thousands of class examples, they cannot be used to directly recognise text regions in photos. These models must be adjusted for the particular issue of separating text from non-text regions. These models' convolutional bases can be trained entirely from scratch or pre-trained models can be improved by giving them practise on regions with and without text.

3.3 Determining Bounding Boxes :

The individual characters will initially be combined into a single connected component before the bounding box of the text region is computed. To eliminate any outliers, this can be done by morphologically closing and then opening.

3.4 Text Extraction :

Text extraction is carried out using the Tesseract OCR package, which consists of an optical character recognition (OCR) engine called libtesseract and a command-line tool named Tesseract. Tesseract now includes a new neural network called the Long Short-Term Memory (LSTM) based OCR engine, which focuses on line identification and also recognises letter pattern. The LSTM network is the foundation of recurrent neural networks. Text is extracted using the Python-Tesseract, an optical character recognition (OCR) programme. This tool can "read" and identify text that is embedded in images. All image types, including JPEG, PNG, GIF, bmp, and others, are read by this programme. It is a wrapper for Google's Tesseract-OCR Engine. The Python Imaging Library is capable of displaying these images. Once the text information has been retrieved from the image files we can utilise it. [3]

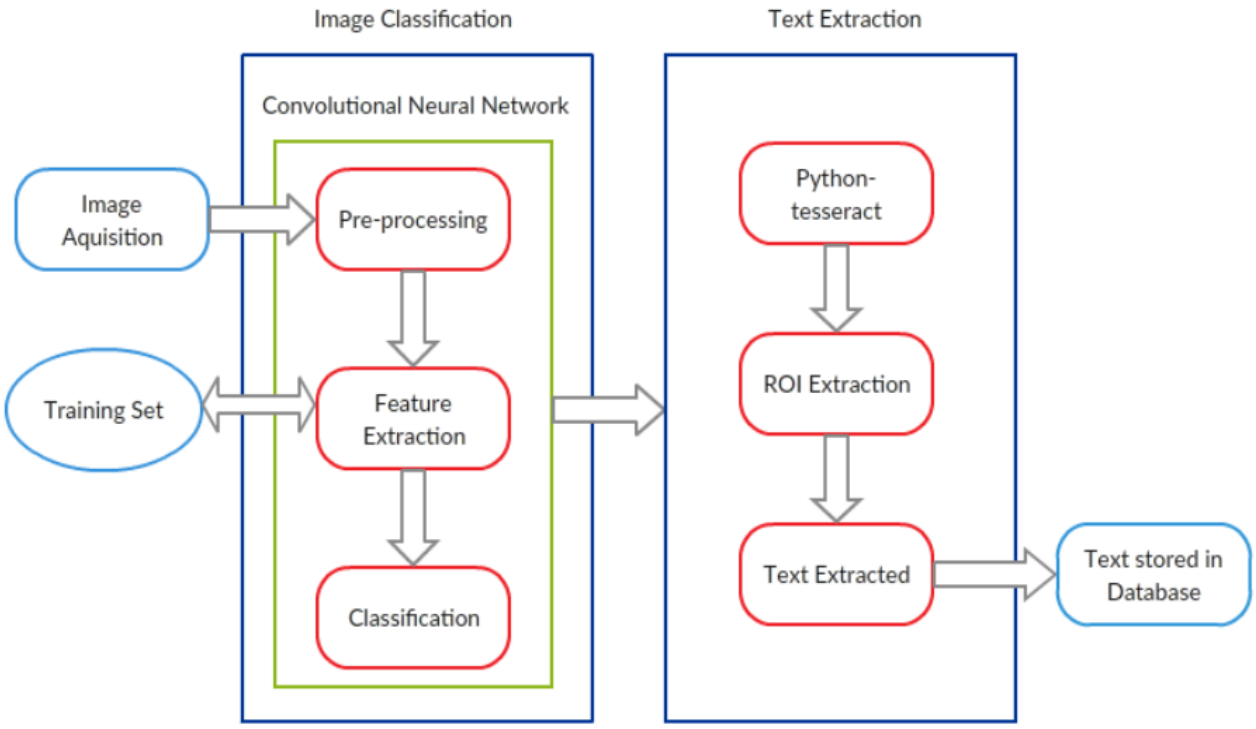
Image Classification
Text Extraction
Convolutional Neural Network
Image Aquisition
Pre-processing
Training Set
Feature Extraction
Classification
Python-tesseract
ROI Extraction
Text Extracted
Text stored in Database

Figure 3.2: [4]

Chapter 4

Description of Deep Learning Model Used

4.1 Optical Character Recognition (OCR)

OCR systems convert a two-dimensional text frame's image or text into text that is machine readable. To do OCR as accurately as is practical, it typically consists of multiple sub-processes.

The sub processes are:

Preprocessing of Image

Localization of the text

Character Recognition

Post Process

4.2 Google Tesseract

The most modern OCR engine,The Google Tesseract provides excellent accuracy in a wide range of languages, including English. In our system, we employed Tesseract version 5.2.0, a recent release from Google. Version 5.2.0 is built using deep neural networks, more especially recurrent neural networks with extended short-term memory architecture. The English text lines are entered into the recognition engine, which then outputs the pertinent textual strings.

In HP Labs, Bristol, Tesseract was first developed as a Ph.D. research project. Between 1984 and 1994, HP created it and it became well-known. Tesseract was made available as open-source software by HP in 2005. It has been created by Google since 2006. [8]

4.3 Tesseract-OCR

Available under the Apache 2.0 licence is the Tesseract-OCR Engine. To extract printed text from photos, use this technique. Many different languages are supported. Tesseract doesn't come with a built-in GUI, but there are a number of them on third-party websites. Tesseract works with a wide range of frameworks and computer languages. It can be used in conjunction with the current layout analysis to identify text within a huge document or with an outside text detector to identify text from a picture of a single text line. [8]

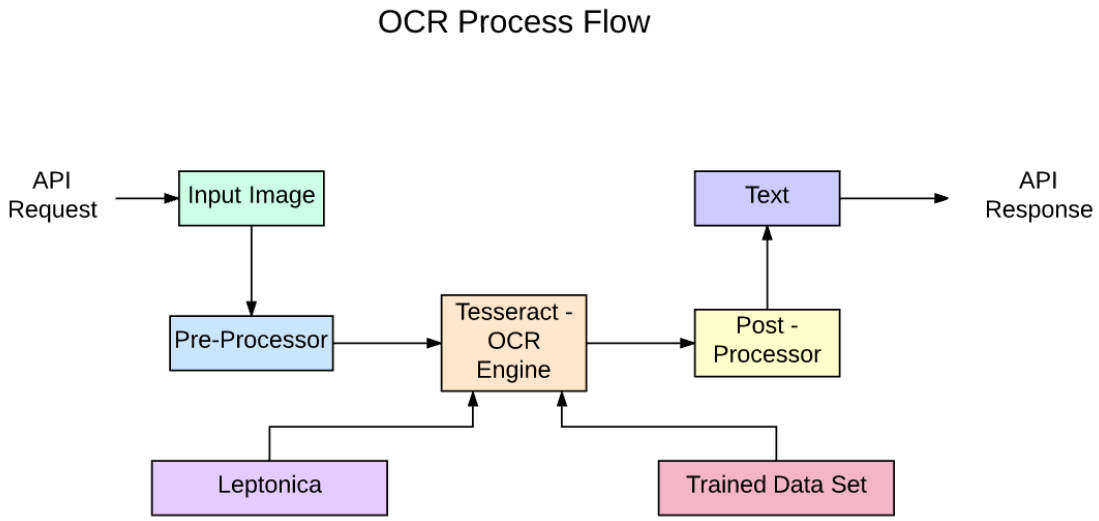

Figure 4.1: [8]OCR Process Flow to build API with Tesseract

OCR Process Flow from a Blog Post to Tesseract API Development Tesseract 4.00 has a new neural network subsystem that is configured as a text line recognizer. It is based on the Python-based LSTM implementation from OCRopus and was adapted for Tesseract in C++. Because TensorFlow also offers a network description language called Variable Graph Specification Language, even though Tesseract's neural network architecture predates TensorFlow, it is compatible with it (VGSL).

Usually, a Convolutional Neural Network is used to recognise an image from a single character (CNN). Text of any length is made up of character sequences, and RNNs—of which LSTM is a well-liked variant—can be used to tackle such issues.

4.4 Convolutional Neural Network (CNN)

In order to accomplish sophisticated image categorization, CNN, a straightforward deep learning-based method, transforms input data into a meaningful representation. As seen in

Figure 4.2, these typically include convolutional layers, pooling layers, and fully linked layers (also known as dense layers). By giving each neuron, which connects to some of the preceding neurons, the proper weights and biases, convolutional layers learn a feature representation of the input image.

An activation function, such as Rectified Linear Unit, is passed the result (ReLU). The size of features is reduced in part by the pooling layer. With the use of probabilities determined by softmax, a fully connected layer combines all the learnt features of the preceding layers to provide predictions in the output layer (classification layer). [9]

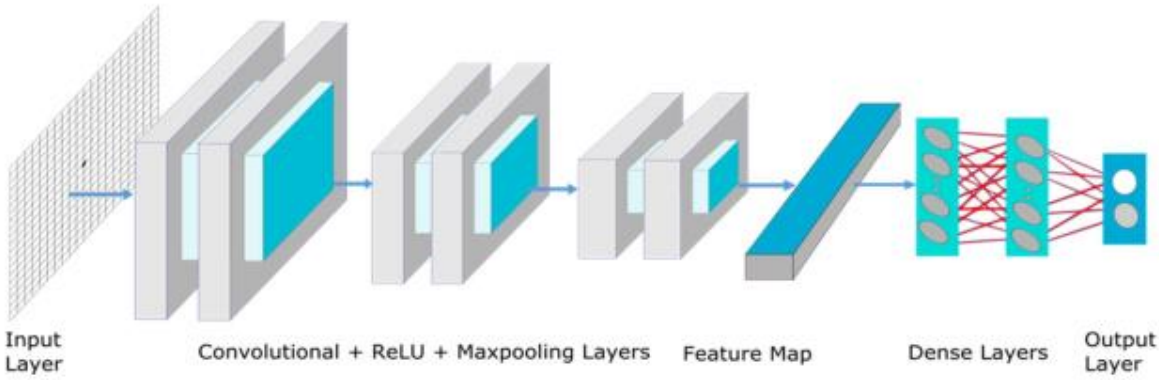

Figure 4.2: [9]

4.5 Long short-term memory (LSTM)

Although LSTMs are great at learning sequences, they become quite slow when there are many states. According to empirical data, giving an LSTM a long sequence to learn is superior to giving it a short sequence with a lot of classes to learn. The Python-based OCRopus model, which was a copy of the C++-based LSMT known as CLSTM, is where Tesseract originated. The CLSTM implementation of the LSTM recurrent neural network makes use of the Eigen toolkit in C++ to do numerical calculations. [8]

Code cleanup and the addition of a new LSTM model were efforts made to modernise the Tesseract tool. Line by line, boxes (rectangles) of the input image are analysed before

being fed into the LSTM model and producing output. We can see how it functions in the figure below.

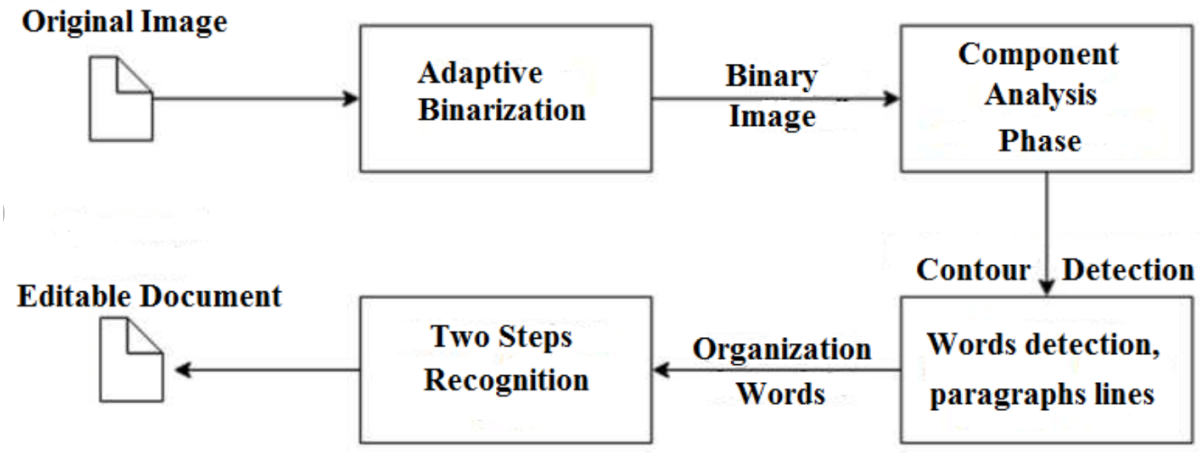

Figure 4.3: Tesseract OCR process [8]

Tesseract 3 OCR process Legacy Tesseract 3.x was dependent on the multi-stage process where we can differentiate steps:

Word finding

Line finding

Character classification

To find words, text lines were combined into blobs, and the lines and regions were inspected for fixed pitch or proportional text. Text lines can be broken up into words in a number of different ways depending on the character spacing style used. Then, the recognition procedure is repeated twice. The first step is to attempt to recognise each word separately. Every term that is accepted is given to an adaptive classifier as training data. The Tesseract tool has been updated through code cleanup and the addition of a new LSTM model. Prior to feeding the input image's boxes (rectangles) into the LSTM model and producing output, each line of the input image is examined. In the below-shown figure 4.4, we can see how it works.

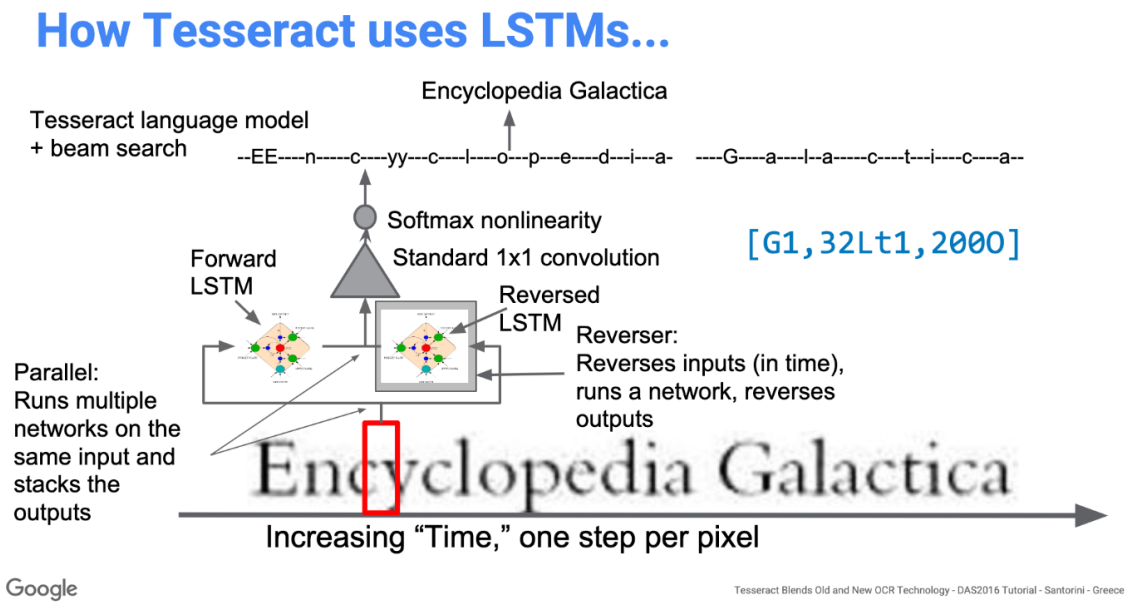

Figure 4.4: How Tesseract uses LSTM model [8]

Tesseract performs better following the addition of a new training tool and extensive data and font training for the model. However, it is still insufficient to work with handwritten writing and unusual fonts. Top layers can be adjusted or retrained for experimentation.

Chapter 5

Results and Conclusions

5.1 Results

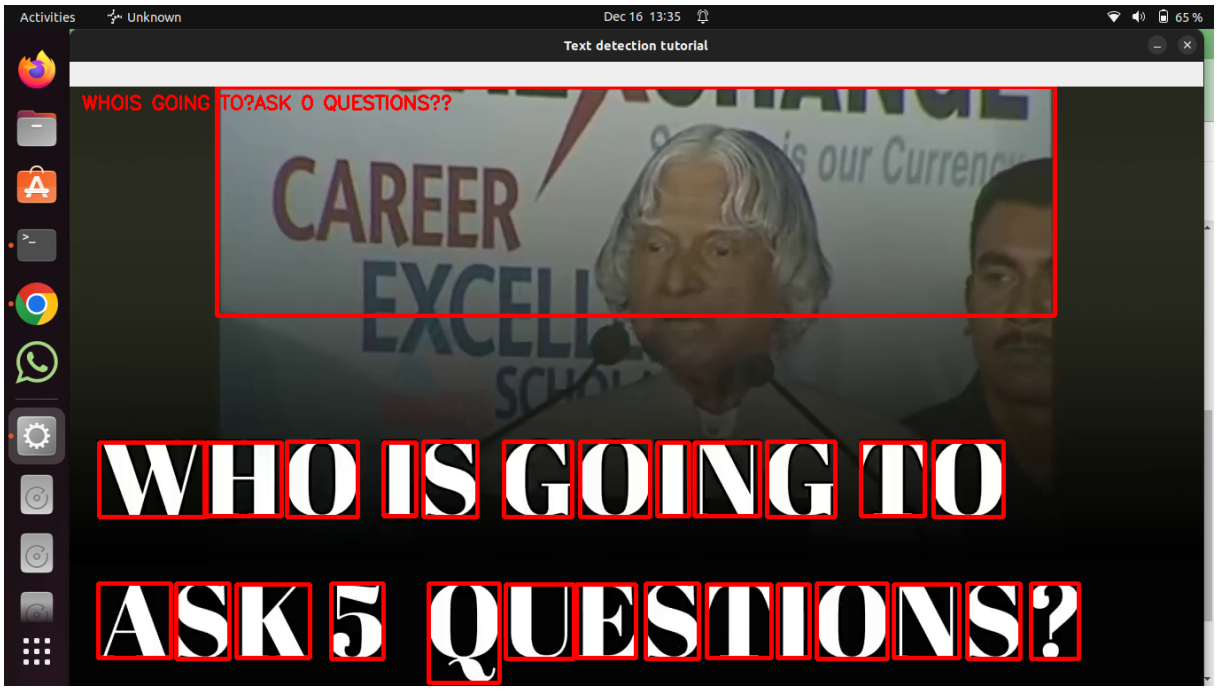

Figure 5.1: Results 1

Figure 5.2: Results 2

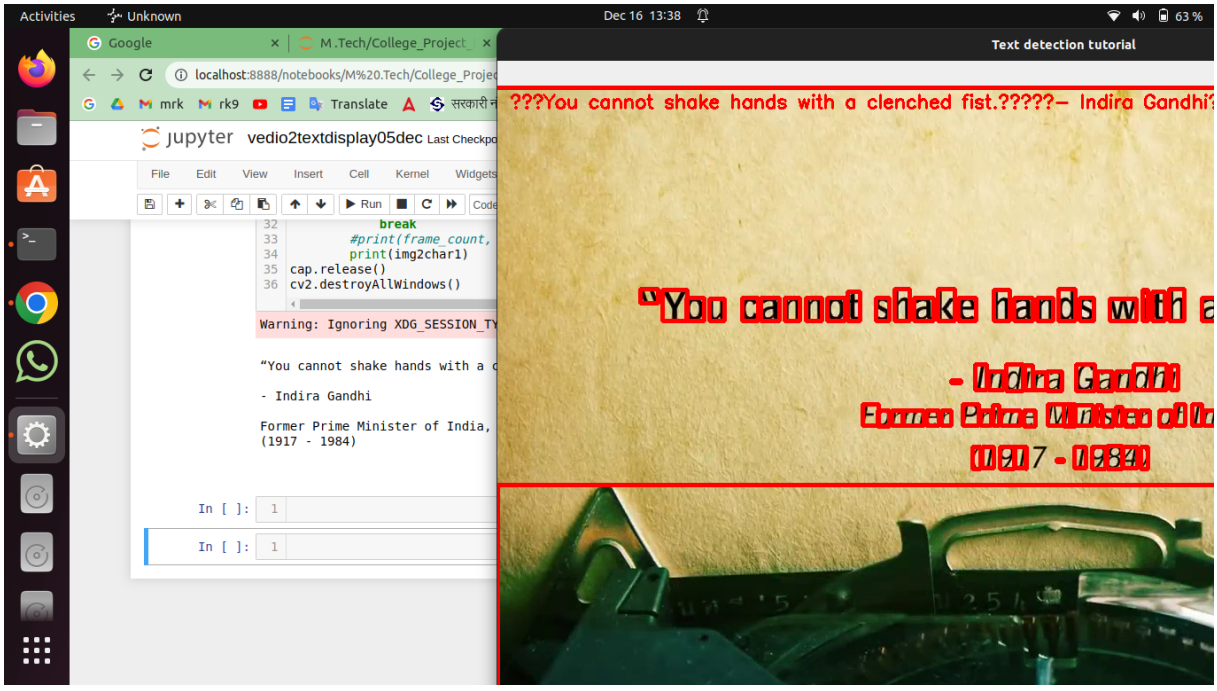

Figure 5.3: Results 3

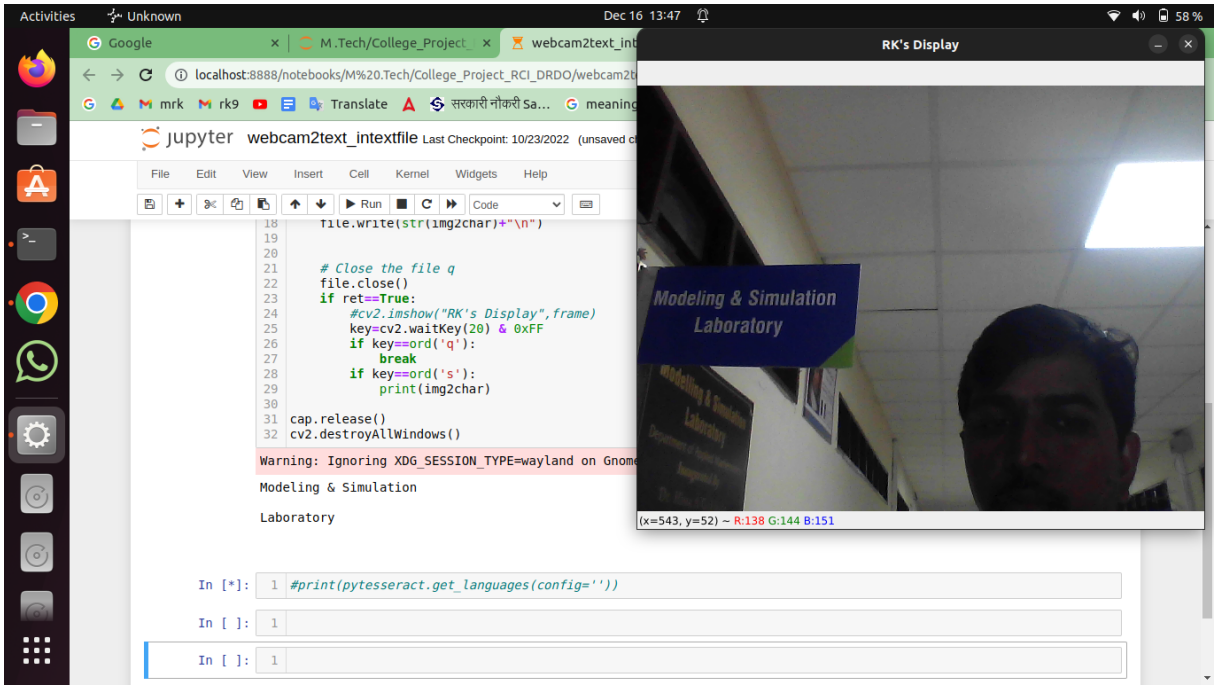

Figure 5.4: Results 4

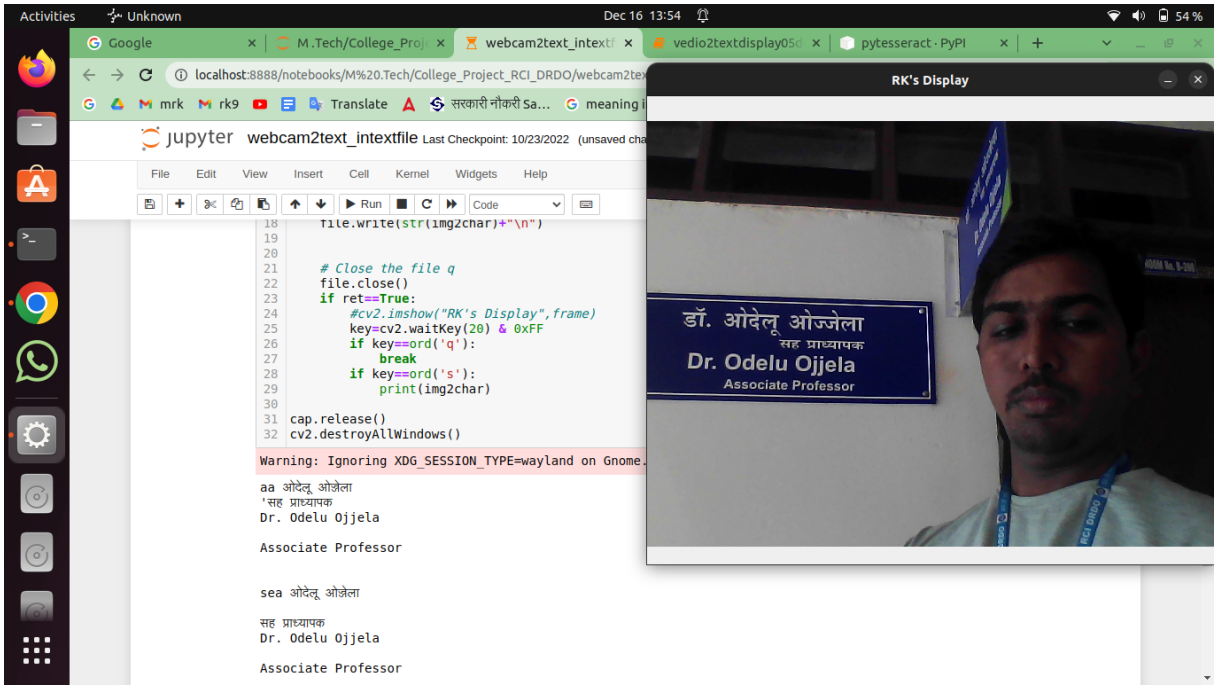

Figure 5.5: Results 5

Figure 5.6: Results 6

5.2 Conclusion

In the current work, we have talked about our suggested approach to identifying and extracting text from video. The technology automates the labor-intensive manual process of extracting text from films, making it time and labor-efficient. Python is the programming language used to implement the system. The technique is mostly utilised for news and instructional videos that include text-based content. In the current study, we propose a thorough framework for English-language text identification and recognition in video frames with text occurrences. We made use of a dataset of video frames that was combined with ground truth data to assess recognition and detection tasks. We developed cutting-edge deep learning-based object detectors and optimised them for text region recognition.We employed cutting-edge deep learning-based object detectors for the recognition of text regions and tweaked them to identify text in various scripts. CNNs are used in a classification framework to identify the script of identified text sections. We used the English Language Net, a CNN and bidirectional LSTM system, which boasts strong recognition rates for difficult

English-language video text.

Modern deep learning-based object detectors that have been calibrated to recognise text in various scripts were employed for the recognition of text regions. The script of detected text regions is recognised using CNNs in a classification framework. The English Language Net, which combines CNN and bidirectional LSTMs, was adopted because it boasts good recognition rates for difficult video text in the English language.

Bibliography

[1] Kiran Agre, Sairaj Gaonkar Ankur Chheda, and Prof. Mahendra Patil. Text recognition and extraction from video, international journal of engineering research technology. page (Volume 5 – Issue 01). IJERT, 2017.

[2] Baseem Bouaziz, Tarek Zlitni, and Walid Mahdi. avitext: Automatic video text extraction; a new approach for video content indexing application. In *2008 3rd International Conference on Information and Communication Technologies: From Theory to Applications*, pages 1–5. IEEE, 2008.

[3] Datong Chen, Jean-Marc Odobez, and Herve Bourlard. Text detection and recognition in images and video frames. *Pattern recognition*, 37(3):595–608, 2004.

[4] R Deepa and Kiran N Lalwani. Image classification and text extraction using machine learning. In *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 680–684. IEEE, 2019.

[5] Lifang Gu. Text detection and extraction in mpeg video sequences. In *Proceedings of the International Workshop on Content-Based Multimedia Indexing*, pages 233–240. Citeseer, 2001.

[6] Anubhav Kumar and Neeta Awasthi. An efficient algorithm for text localization and extraction in complex video text images. In *2013 2nd International Conference on Information Management in the Knowledge Economy*, pages 14–19. IEEE, 2013.

[7] Punith Kumar and PS Puttaswamy. Moving text line detection and extraction in tv video frames. In *2015 IEEE International Advance Computing Conference (IACC)*, pages 24–28. IEEE, 2015.

[8] nano notes. Tesseract, opencv and python, 1999.

[9] Jawad Rasheed, Hasibe Busra Dogru, and Akhtar Jamil. Turkish text detection system from videos using machine learning and deep learning techniques. In *2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP)*, pages 116–120. IEEE, 2020.