Rajiv Gandhi University of Knowledge Technologies

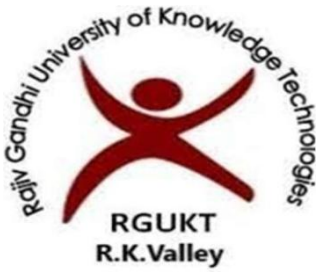
R.K Valley, Y.S.R Kadapa (Dist)-516330

A
project report on

TEXT DETECTION AND RECOGNITION IN IMAGES

Submitted by

P. Naga Hari Chandana R171028



Under the esteemed guidance of

Mr. Penugonda Ravi Kumar
(Asst Professor)

Department of Computer Science Engineering

This project report has been submitted in fulfillment of the requirements for the Degree of Bachelor of Technology in software Engineering.

April - 2023

Rajiv Gandhi University of Knowledge Technologies
IIIT, R. K. Valley, YSR Kadapa (Dist) -516330

CERTIFICATE

This is to certify that report entitled "Text Detection and Recognition in Images" Submitted P.Naga Hari Cahandan (R171028) in partial fulfillment of the requirements of the award of bachelor of technology in computer science engineering is a bona fide work carried by her under the supervision and guidance.

The report has been not submitted previously in part or full to this or any other university or institute for the award of any degree or diploma.

GUIDE **HEAD OF THE**

DEPARTMENT

Mr. P. Ravi Kumar
Asst .Prof in Dept of CSE
RGUKT-RK Valley.

Mr. N. Satyanandaram
HOD of CSE Dept
RGUKT-RKValley.

Submitted for the practical examination held on…….

Internal Examiner External Examiner

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and who's constant guidance and encouragement crown all the efforts success.

I would like to express my sincere gratitude to Mr. P. Ravi Kumar, my project guide for valuable suggestions and keen interest throughout the progress of my project.

I'm grateful to Mr. N. Satyanandaram HOD CSE, for providing excellent computing facilities and congenial atmosphere for progressing my project.

At the outset, I would like to thank Rajiv Gandhi University of Knowledge Technologies (RGUKT), for providing all the necessary resources and support for the successful completion of my course work.

DECLARATION

I hereby declare that this report entitled “Text Detection and Recognition in Images” Submitted by us under the guidance and supervision of Mr. P. Ravi Kumar, is a bona fide work. I also declare that it has not been of Submitted previously in part or in full to this University or other institution for the award of any degree or diploma.

Date: 24-04-2023
Place: - RK Valley

P. Naga Hari Chandana(R171028)

TABLE OF CONTENTS

ABSTRACT

Content-based indexing and retrieval systems might benefit from text characters or forms embedded in photographs or images as a source of data.These texts, however, due to a variety of factors such as relative sizes, grayscale values, and complicated background patterns, these texts are difficult to detect and distinguish. The strategies for developing an effective application system for identifying and recognizing text embedded in images are discussed in this research. Both empirical image processing methods and statistical machine learning and modeling approaches are studied in two sub-problems: text detection or identification and text recognition.

Using machine learning methods for text detection would be difficult for a variety of reasons. To address these issues, we propose a two-step localization/verification procedure. The first step is to quickly localize candidate text lines, allowing for character normalization into a single size. To remove useless data, a trained support vector machine or multi-layer perceptron is applied on background independent features during the verification step.

Text detection is accomplished in two steps, combining the speed of a text localization step, which allows for text size normalization, with the strength of a machine learning text verification step applied on background independent features. Text recognition is addressed through a text segmentation step followed by a traditional OCR algorithm within a multi-hypotheses framework based on multiple segments, language modelling, and OCR statistics.

INTRODUCTION

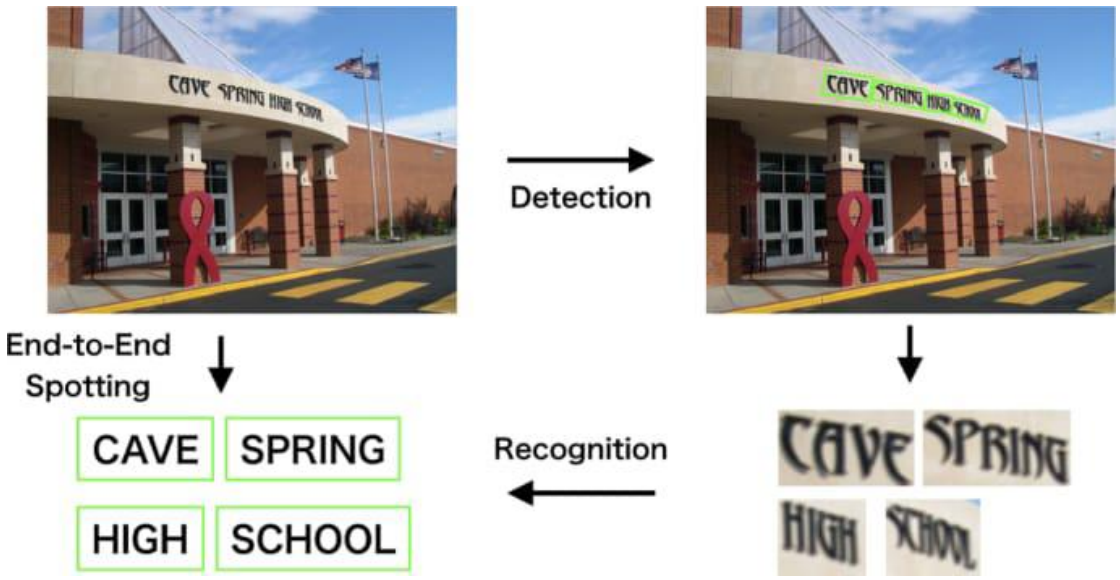
Text detection and recognition in images is a research area which attempts to develop a computer system with the ability to automatically read from images or any bundle of images the text content visually embedded in complex backgrounds. That should answer two typical questions “Where & What”: “where is a text string?” and “what does the text string say?” in an image. In other words, using such a system, text embedded in complex backgrounds can be automatically detected and each character or word can be recognized.

Content-based multimedia database indexing and retrieval tasks require automatic extraction of descriptive features that are relevant to the subject materials (images, video, etc.). The typical low-level features that are extracted in images and video include measures of color, texture, or shape. Although these features can easily be obtained, they do not give a precise idea of the image content. Extracting more descriptive features and higher level entities, such as text and human faces, has recently attracted significant research interest. Text embedded in images and video, especially captions, provide brief and important content information, such as the name of players or speakers, the title, location, date of an event, etc.

Text detection and recognition in images, which aims at integrating advanced optical character recognition (OCR) and text-based searching technologies, is now recognized as a key component in the development of advanced image and video annotation and retrieval systems. Unfortunately, text characters contained in images and videos can be any gray-scale value (not always white), low-resolution, variable size and embedded in complex backgrounds.

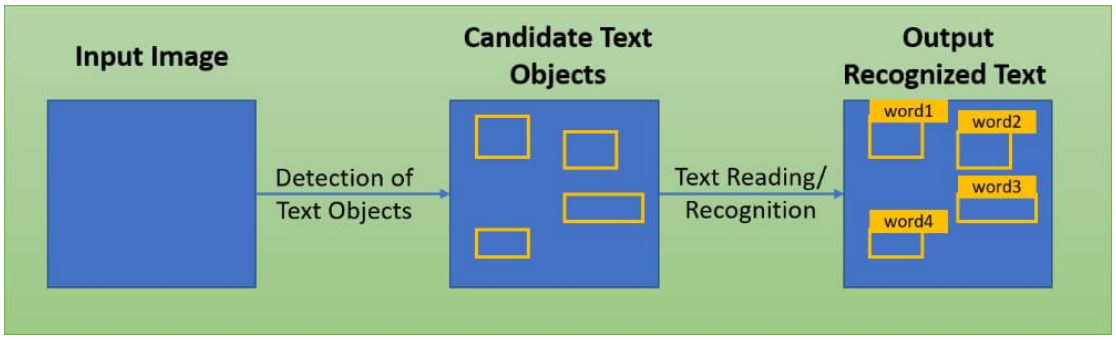
Literature Review

As researched by Ouais Alsharif and Joelle Pineau proposed an hybrid HMM Maxout Models to recognize the text, characters and words. In their paper, they focused on using leveraged convolutional Maxout networks to outperform the state-of-the-art in character recognition, demonstrating how to use the character recognizer in a word recognizer that is fast, tunable, highly accurate, and scales gracefully with lexicon size. They built an end-to-end text recognition system using the previous modules proposed previously, as well as other, relatively simple constructs. As there focus totally on the end to end parts of the text recognition process that depends mostly on the image resolution, gray scale, and the clarity and the quantity of the images.In there observation has a 0.5 error rate and 60 percent of the accuracy (depends on the images) which predicted by convolutional Maxout networks.



As researched by Dr. M Meena kumari ,Dr. T. Mohanasundaram and R Suresh Kumar proposed an relevant approach in their paper named An Efficient Method for Text Detection and Recognition, their results demonstrated that the new adaptive algorithm could efficiently recognize the text even if the image is corrupted by high noise density. Also implementation of the proposed method is easy and less complex when compared to other methods, that shows how to split regions containing text in an image and how to detect the text. The automated text detection algorithm uses MSER regions and optical character recognition (OCR) is used to recognize the text. The image first converts in to different grey scales, black&white and change

the pixels into red green blue RGB format and store that format and get recognized by the algorithm. In there observation has a 0.5 error rate and 80 percent of the accuracy without any effect of what type of image is that which predicted by optical character recognition process.



As researched by Amritha S Nadarajan&; Thamizharasi A (2018), proposes an innovative algorithm to find value of stroke width in natural images. This algorithm helps to detect many font and languages. This includes preprocessing, extraction or text localization, classification and character detection.In their paper they provides a detailed study of evolution of text detection in natural images. It compares, analyzes and also discusses the different methods to overcome existing challenges in text detection. In their comparative study proves that CNN is a better technique to detect text in natural images. The Convolutional neural networks uses different features into different states and connects every neuron and analysis the effect of every unique pixel and and get the accuracy to be less error rate.

As researched by Datong chen proposed a OCR software and a selecction algorithm based on on language modeling and OCR statistics chooses the text result from all the produced text strings. After segmentation there used a ROVER (Recognizer Output Voting Error Reduction) algorithm is studied for improving the final recognition text string. In this paper there observed results are less error rate and more than 80 percent accuracy is detected.

Problem Identification & Objectives

For a Image the objective is to detect the textual region by plotting the bounding box and after that, the detected text has to be recognized.Texts in images can be in different language styles, colors, fonts, sizes, orientations, and shapes. We have to deal with these texts in natural scene images that exhibit higher diversity and variability.Images may have backgrounds with patterns or objects with a shape that is extremely similar to any text which creates problems while detecting texts.Disrupted images (low quality/resolution/multi-orientation).Low latency is required to detect, recognize and translate the text in the images in real-time.

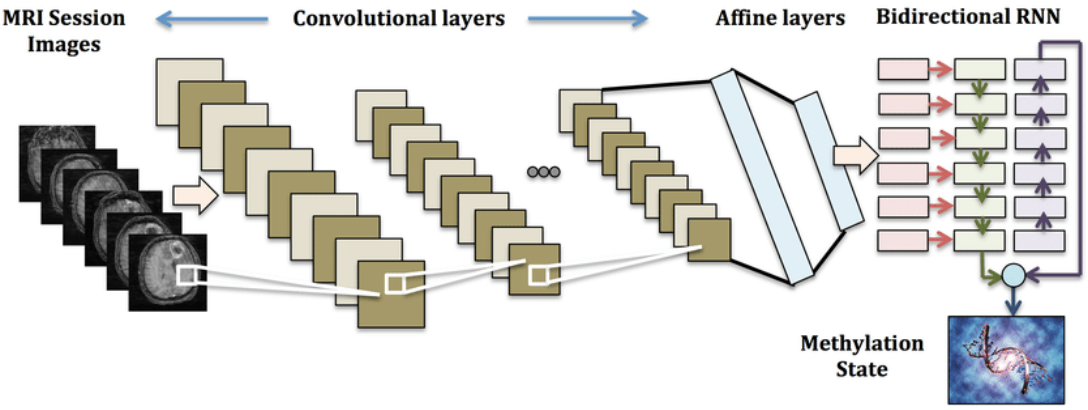
In the problem caused by the different styles, fonts and sizes of images, the main conflict arises when the detection of the boarders and edges of the detected text should be very accurate. If there any mistake is done in the analysis by the algorithm that we used the results creates a drastic different in the characters accuracy.

Existing Model

In other existing model uses CNN, RNN and also developed an attention-based spatial transformer network (STN) to rectifying text distortion for recognition and the detection of the text in images.

In this the images need to be analyzed through with many of the convolutional layers that are added by the training data for the detection process. Convolutional neural networks is a classification of the data using a mechanism known as filters, followed by pooling layers. A

filter is a matrix of randomized numbers. In a CNN, filters are multiplied against matrix representations of parts of the image, effectively scanning the picture pixel by pixel and getting the average value of all adjacent pixels, thereby detecting the most important features

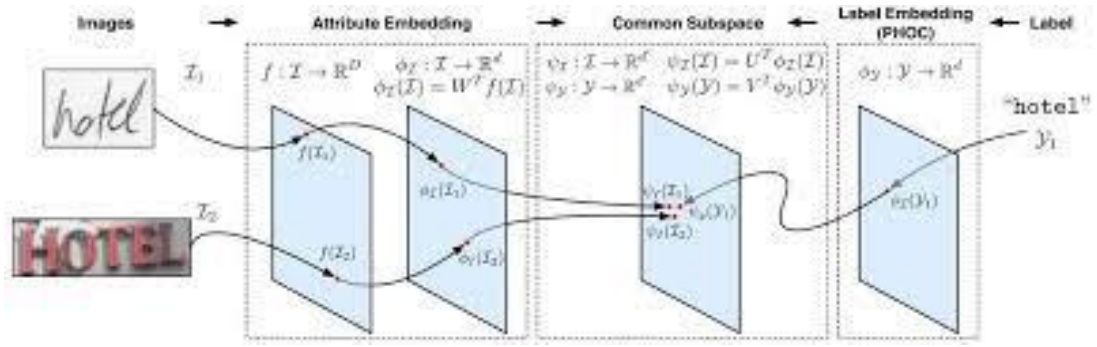


In this bidirectional recurrent neural networks(RNN) are used because they are really just putting two independent recurrent neural networks RNNs together. The input sequence is fed in normal time order for one network, and in reverse time order for another. The outputs of the two networks are usually concatenated at each time step, though there are other option to be analyzed.

Proposed Model

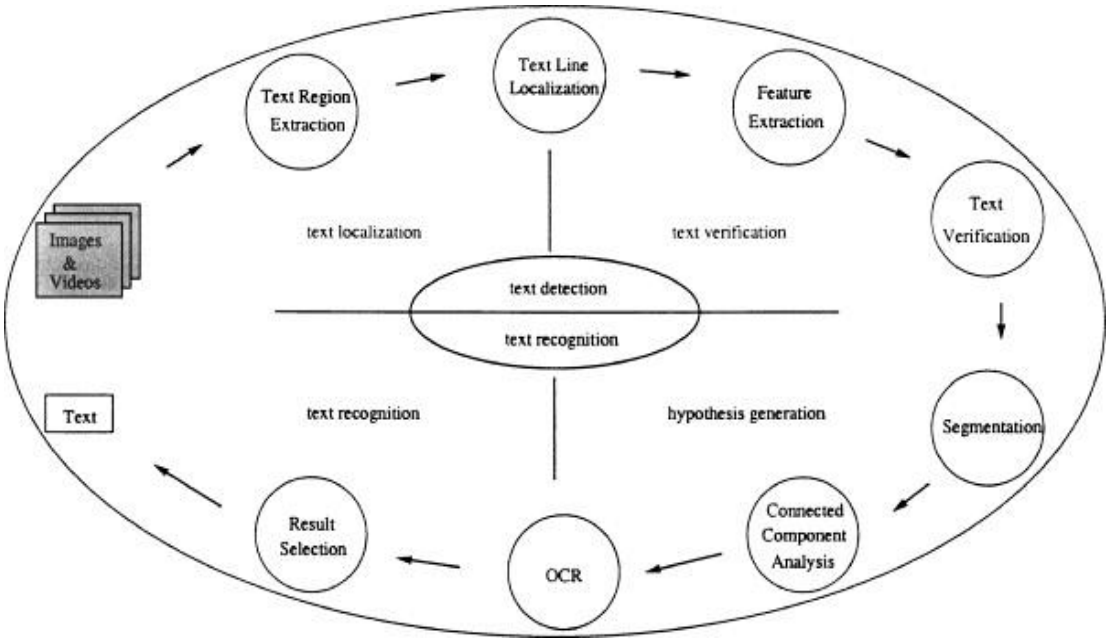
In our proposed model we use OCR algorithm to recognize and detect the text in images that may be blurred, low resolution or any curved images with a good amount accuracy.

The method we propose belongs to the top-down category, and consists of two main tasks as text regions. Following the cascade filtering idea, which consists of the sequential processing of data with more and more selective filters, the text detection task is decomposed into two sub tasks. These are a text localization step, whose goal is to quickly extract potential text blocks in images with a very low missing rate and a reasonable false alarm rate, and a text verification step based on a powerful machine learning tool. Such an approach allows to obtain high performance with a lower computational cost in comparison to other methods.



System Methodology

Proposed Process



Steps to be follow:

* Take a input image containing text,

* Firstly extract the text region from the image by text line localization,

* Verify the text that is correctly extracted or not,

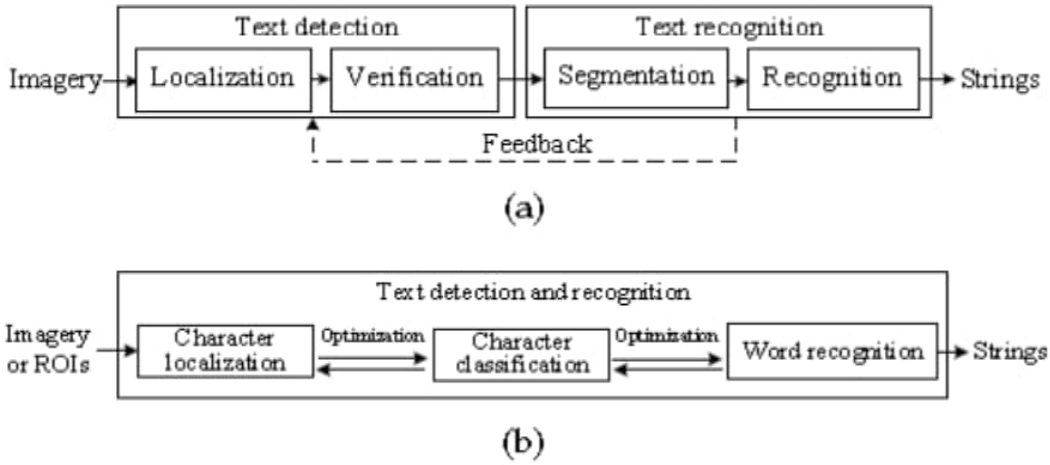
* Now generate a segment hypothesis,

* By using OCR recognize the text.

Text Detection and Localization the approach for processing is used Connected Components analysis, Region based methods and Supervised Approach.

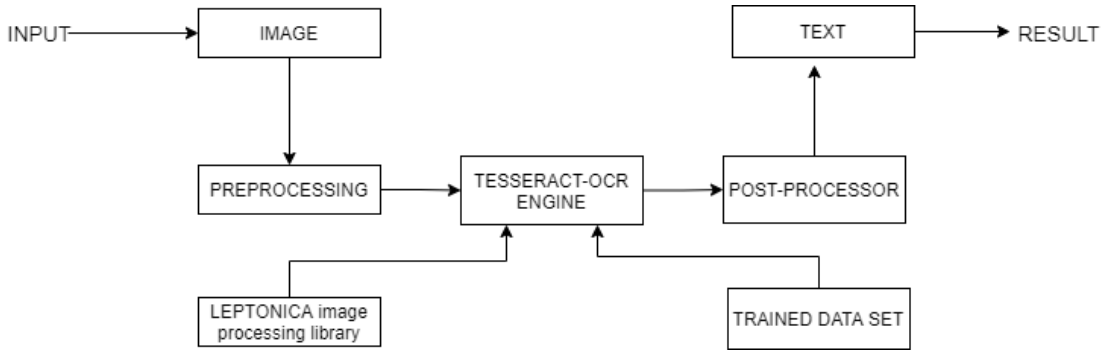
The Connected Components analysis has vast features like Graph based method High speed, Uses color or edge features , Not efficient for noisy images. Region based methods has been used because of its unique nature are Windowing based approach, the lesser speed, the Use texture features or morphological operations, the Efficient for noisy images also. And the

Supervised Approach features are Supervised classifier has training phase Classifier knows features of the text before classification starts.



Tesseract-ocr flow

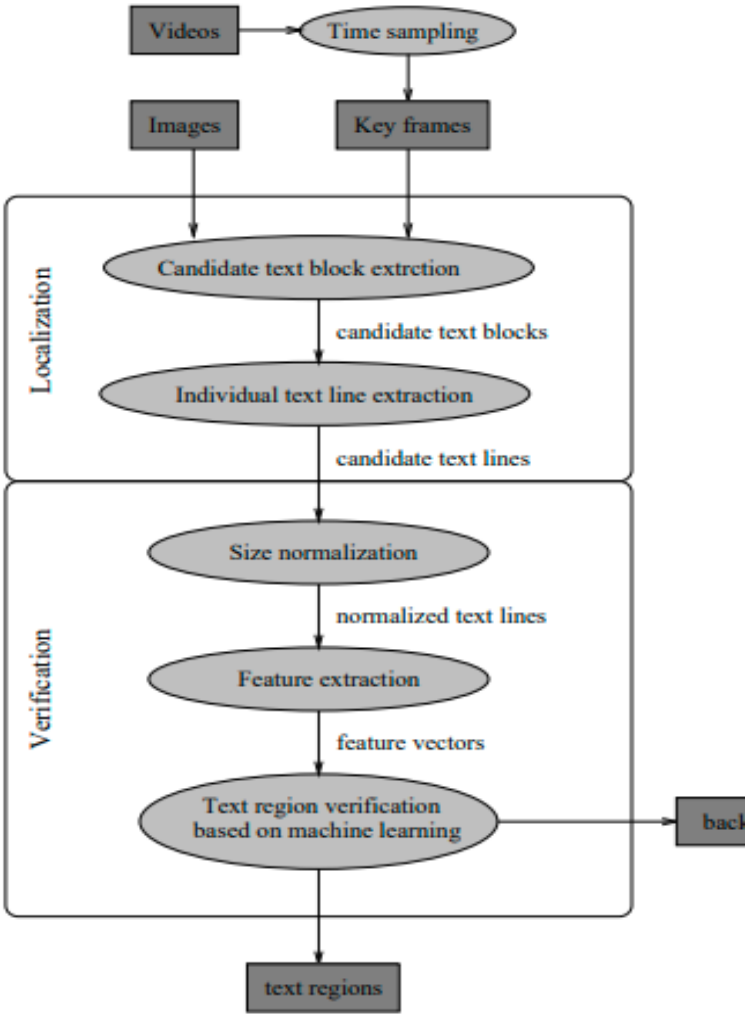
Tesseract is an open source optical character recognition (OCR) platform. OCR extracts text from images and documents without a text layer and outputs the document into a new searchable text file, PDF, or most other popular formats. Tesseract is highly customizable and can operate using most languages, including multilingual documents and vertical text. Although the software can be used on Windows or Linux, this guide will be based on Mac operating systems which is done through the terminal application.



Text Recognition is of two different parts they are character recognition and word recognition. The character recognition is to divide text into cut-outs of single characters and Independent of lexicon is should be organized, also used when number of words to be recognized are not limited and the word recognition that identifies word from text image then recognizes small

number of words provided by lexicon, it is suitable only for recognizing limited number of words

In the localization/detection the first step locates candidate text blocks in images with a fast algorithm. This localization process avoids applying the machine learning classifiers on the whole images as well as to further reduce the variation of text size by extracting individual text strings (lines). To obtain a fast algorithm, candidate text blocks are located by exploring heuristic characteristics. We use a threshold in this algorithm to adjust the weakness of the heuristic feature based classifiers in distinguishing text and backgrounds. A low threshold is useful to avoid rejecting text blocks (low rejection rate). The resulting false alarms will be removed in the following verification step.

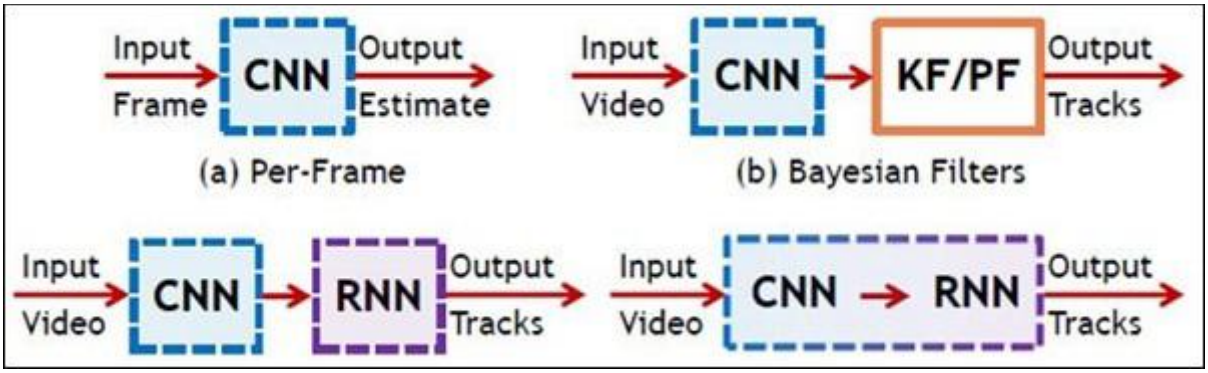


CNN:

Convolutional Neural Networks, or CNNs, were designed to map image data to an output variable.They have proven so effective that they are the go-to method for any type of prediction problem involving image data as an input.

The benefit of using CNNs is their ability to develop an internal representation of a two-dimensional image. This allows the model to learn position and scale in variant structures in the data, which is important when working with images.

The CNN input is traditionally two-dimensional, a field or matrix, but can also be changed to be one-dimensional, allowing it to develop an internal representation of a one-dimensional sequence. This allows the CNN to be used more generally on other types of data that has a spatial relationship. For example, there is an order relationship between words in a document of text. There is an ordered relationship in the time steps of a time series.

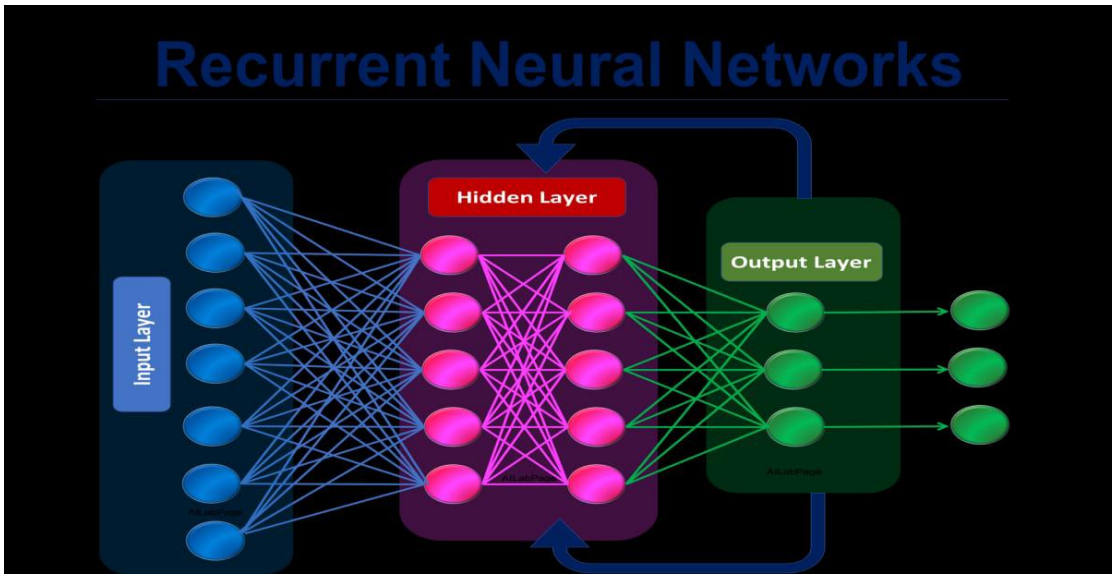


RNN

RNNs are designed to make use of sequential data, when the current step has some kind of relation with the previous steps. This makes them ideal for applications with a time component (audio, time-series data) and natural language processing. RNN's perform very well for applications where sequential information is clearly important, because the meaning could be misinterpreted or the grammar could be incorrect if sequential information is not used. Applications include image captioning, language modeling and machine translation.

In recurrent networks, history is represented by neurons with recurrent connections - history length is unlimited. Also, recurrent networks can learn to compress whole

history in low dimensional space, while feedforward networks compress (project) just single word. Recurrent networks have possibility to form short term memory, so they can better deal with position invariance feedforward networks cannot do that.



Overview of Technologies

Python:

Python is an interpreted high-level general-purpose programming language. With its use of significant indentation, its design philosophy emphasis code readability. Its language constructs and object-oriented approach are intended to assist programmers in writing clear, logical code for small and large-scale projects.

Python is garbage-collected and dynamically typed. It supports a wide range of programming paradigms, including structured (especially procedural), object-oriented, and functional programming. Because of its extensive standard library, it is frequently referred to as a "batteries included" language.

Features of Python:

- Easy to Learn and Use: Python is easy to learn and use. It is developer-friendly and high level programming language.
- Expressive Language: Python language is more expressive means that it is more understandable and readable.
- Interpreted Language: Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.
- Cross-platform Language: Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language.
- Free and Open Source: Python language is freely available at offical web address.The source-code is also available. Therefore it is open source.
- Object-Oriented Language: Python supports object oriented language and concepts of classes and objects come into existence.
- Extensible: It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.
- Large Standard Library :Python has a large and broad library and prvides rich set of module and functions for rapid application development.

- GUI Programming Support: Graphical user interfaces can be developed using Python.
  Integrated: It can be easily integrated with languages like C, C++, JAVA etc.

Machine learning:

Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

OCR:

Optical character recognition (OCR) technology is a business solution for automating data extraction from printed or written text from a scanned document or image file and then converting the text into a machine-readable form to be used for data processing like editing or searching.

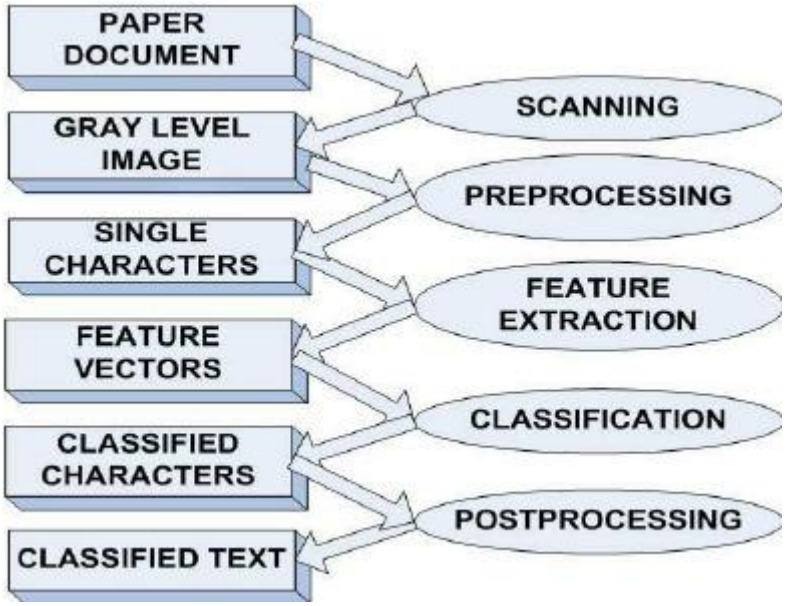
The benefits of OCR technology to businesses include:

- Elimination of manual data entry
- Resource savings due to the ability to process more data faster and with fewer resources

- Error reductions
- Reallocation of physical storage space
- Improved productivity

Businesses that employ OCR capabilities to convert images and PDFs (typically originating as scanned paper documents) save time and resources that would otherwise be necessary to manage unsearchable data. Once transferred, OCR-processed textual information can be used by businesses more easily and quickly

It is a widespread technology to recognize text inside images, such as scanned documents and photos. OCR technology is used to convert virtually any kind of image containing written text (typed, handwritten, or printed) into machine-readable text data.



OCR Algorithm to be followed

Image Acquisition

In this process images of paper documents or any other sources are taken. This way, an original image can be captured and stored in the form of pixels.

Preprocessing

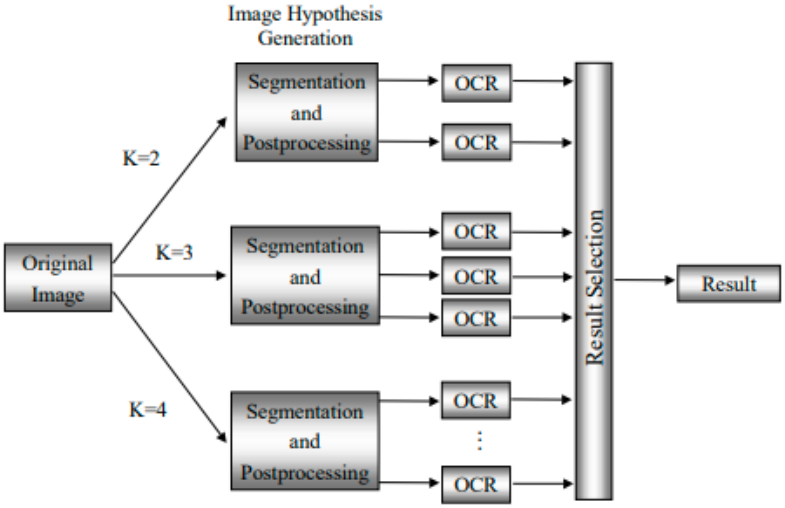
In this the noise level on an image should be optimized and areas outside the text removed

Segmentation

In this process the characters grouped into meaningful chunks. There can be predefined classes for characters. So, images can be scanned for patterns that match the classes

Feature Extraction

Here splitting of input data into a set of features, that is, to find essential characteristics that make one or another pattern recognizable



Training a Neural Network

A training dataset and the methods applied to achieve the best output will depend on a problem that requires an OCR-based solution.

Post-Processing

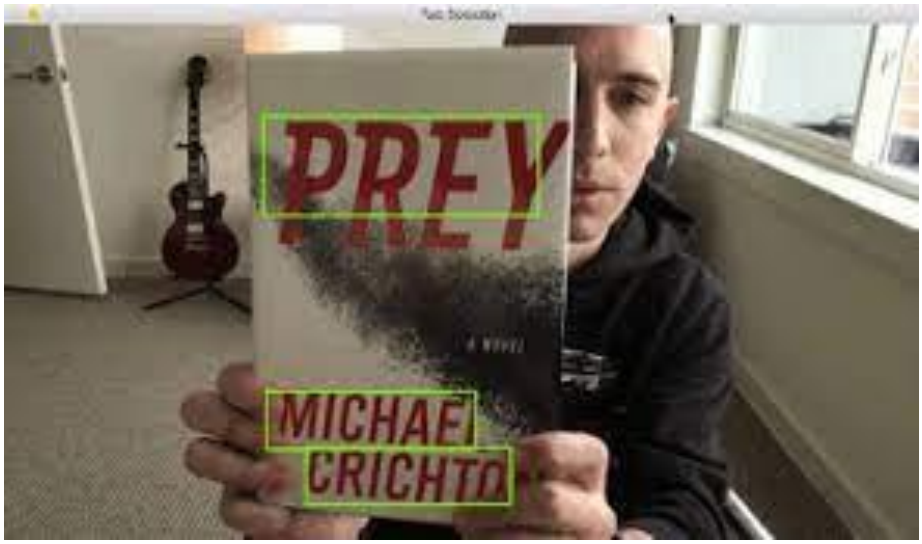
In this step the process of refinement as an OCR model can require some corrections

OpenCV:

OpenCV is a cross-platform library using which we can develop real-time **computer vision applications**. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection

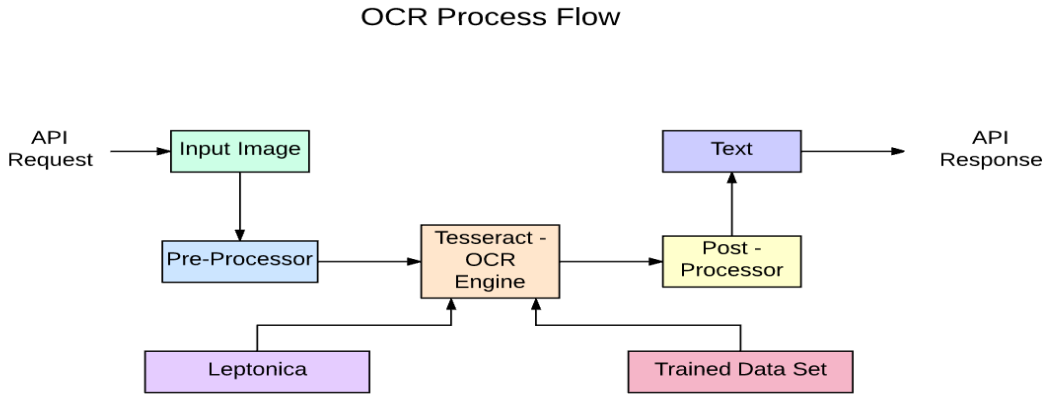
OpenCV is a library of Python bindings that is generally used to solve problems related to computer vision. It is a cross-platform library available in a wide variety of programming languages, such as C++, Python, Java, etc.

OpenCV can be used to process images and videos to identify objects, faces, or even the handwriting of a human. We use vector space and perform mathematical operations on these features to identify image pattern and its various features.



Pytesseract:

Python-tesseract is an optical character recognition (OCR) tool for python. That is, it will recognize and "read" the text embedded in images.Python-tesseract is a wrapper for Google's Tesseract-OCR Engine. It is also useful as a stand-alone invocation script to tesseract, as it can read all image types supported by the Pillow and Leptonica imaging libraries, including jpeg, png, gif, bmp, tiff, and others. Additionally, if used as a script, Python-tesseract will print the recognized text instead of writing it to a file.



Implementation:

Coding:

```
import pytesseract
import cv2
import matplotlib.pyplot as plt
pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'
img=cv2.imread('image.png')
img=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
#plt.imshow(img)
#text = pytesseract.image_to_string(img, lang = 'eng')
#print(text)


#t1=pytesseract.image_to_boxes(img, lang = 'eng')
#print(t1)
"""
hImg,wImg,_=img.shape
boxes = pytesseract.image_to_boxes(img)
for b in boxes.splitlines():
    #print(b)
    b=b.split(' ')
    #print(b)
    x,y,w,h = int(b[1]),int(b[2]),int(b[3]),int(b[4])
    cv2.rectangle(img,(x,hImg-y),(w,hImg-h),(0,255,255),2)
    cv2.putText(img,b[0],(x,hImg-y+25),cv2.FONT_HERSHEY_COMPLEX,1,(50,50,255),2)
"""
```

```
hImg,wImg,_=img.shape
boxes = pytesseract.image_to_data(img)
for t,b in enumerate(boxes.splitlines()):
    if t!=0:
        #print(b)
        b=b.split()
        #print(b)
        if len(b)==12:
            x,y,w,h = int(b[6]),int(b[7]),int(b[8]),int(b[9])
            cv2.rectangle(img,(x,y),(w+x,h+y),(50,50,255),2)
            cv2.putText(img,b[11],(x,y-5),cv2.FONT_HERSHEY_COMPLEX,0.5,(0,255,255),2)
fonT_scale = 1.5
font = cv2.FONT_HERSHEY_PLAIN

cap = cv2.VideoCapture("testvideo12.mp4")

if not cap.isOpened():
    cap = cv2.VideoCapture(0)
if not cap.isOpened():
    raise IOError("Cannot open video")

cntr=0
while True:
    ret,frame = cap.read()
    cntr=cntr+1
    if cntr%20==0:
```

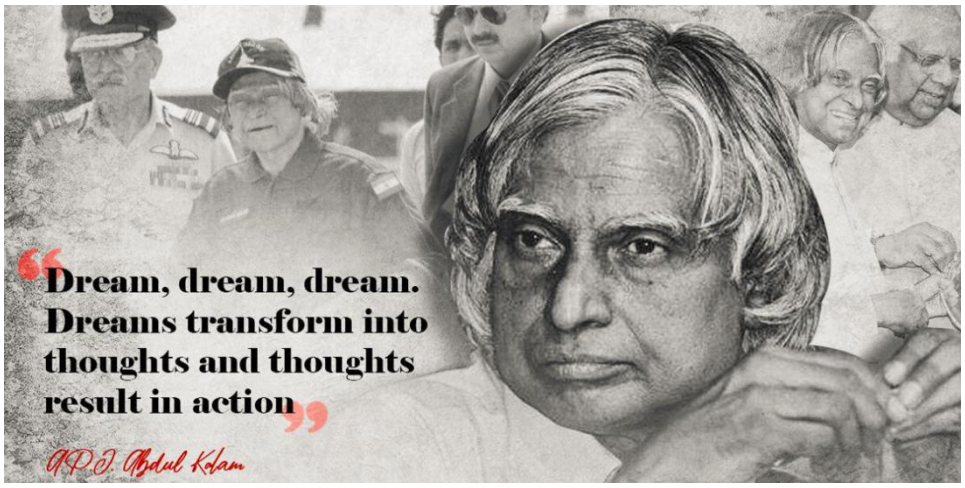
```
        try:
            imgH,imgW,_=frame.shape
        except:
            exit()
        x1,y1,w1,h1=0,0,imgH,imgW
        imgchar = pytesseract.image_to_string(frame, lang = 'eng')

        '''
        imgboxes=pytesseract.image_to_boxes(frame, lang = 'eng')
        for boxes in imgboxes.splitlines():
            boxes = boxes.split(' ')
            x,y,w,h = int(boxes[1]),int(boxes[2]),int(boxes[3]),int(boxes[4])
            cv2.rectangle(frame,(x,imgH-y),(w,imgH-h),(0,0,255),3)
            cv2.putText(frame,boxes[0],(x,imgH-
    y+25),cv2.FONT_HERSHEY_COMPLEX,1,(50,50,255),2)
        '''
        imgboxes=pytesseract.image_to_data(frame, lang = 'eng')
        for box,boxes in enumerate(imgboxes.splitlines()):
            if box!=0:
                boxes = boxes.split()
                if(len(boxes)==12):
                    x,y,w,h = int(boxes[6]),int(boxes[7]),int(boxes[8]),int(boxes[9])
                    cv2.rectangle(frame,(x,y),(w+x,h+y),(0,0,255),3)
                    cv2.putText(frame,boxes[11],(x,y-
    10),cv2.FONT_HERSHEY_COMPLEX,1,(50,50,255),2)
```

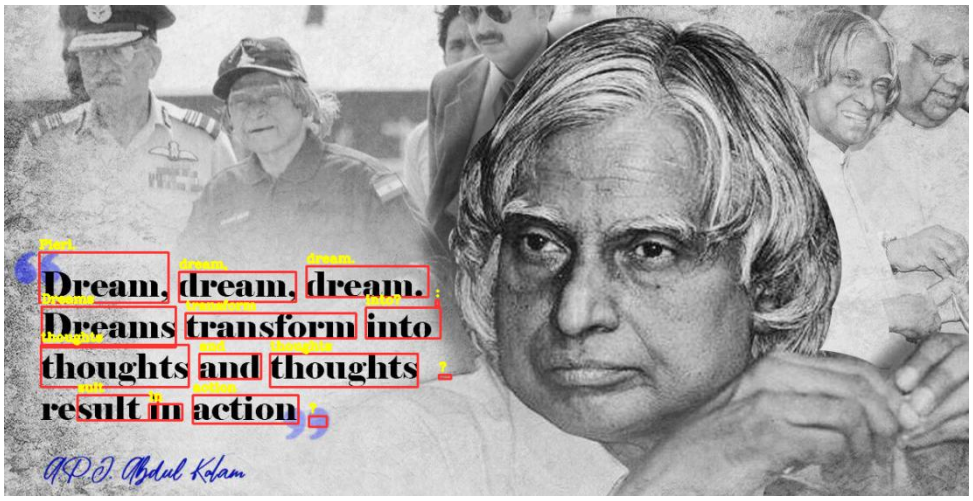
```
#cv2.putText(frame,imgchar,(x1+int(w1/50),y1+int(h1/50)),cv2.FONT_HERSHEY_SIMPLEX,
0.7,(0,0,255),2)
        #font=cv2.FONT_HERSHEY_SIMPLEX
        cv2.imshow("Result",frame)
        print(cv2.waitKey(2))
        print(0xFF == ord('q'))
        if(cv2.waitKey(2)and 0xFF == ord('q')):
            cap.release()
            cv2.destroyAllWindows()
            break
cap.release()
cv2.destroyAllWindows()
cv2.imshow("Result",img)
cv2.waitKey(0)
```

Testing:

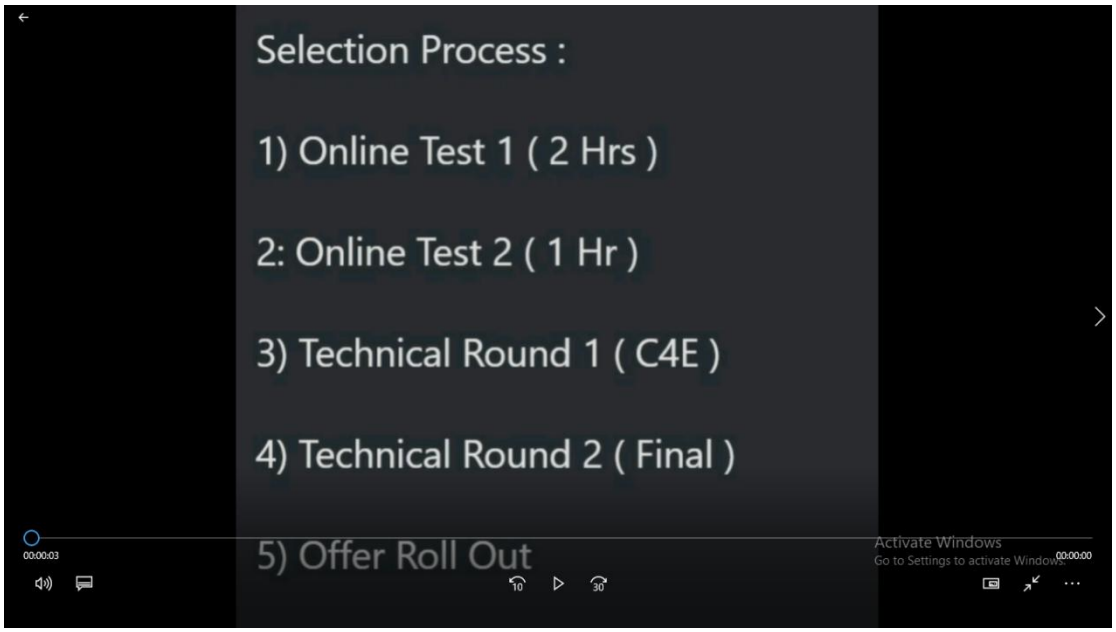
Input image1:



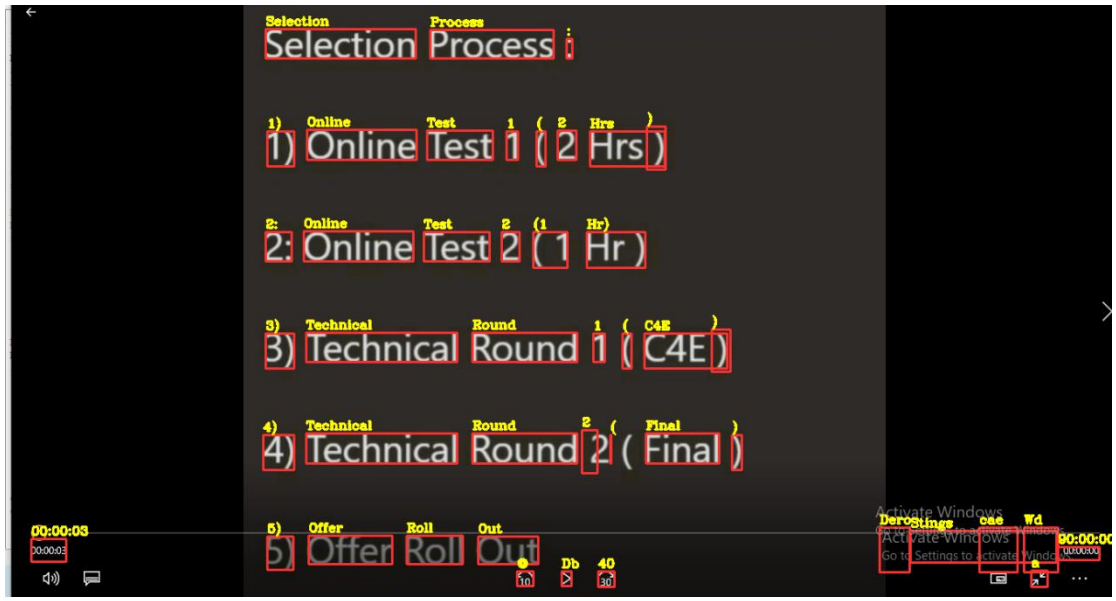
Output1:



Input image2:



Output2:



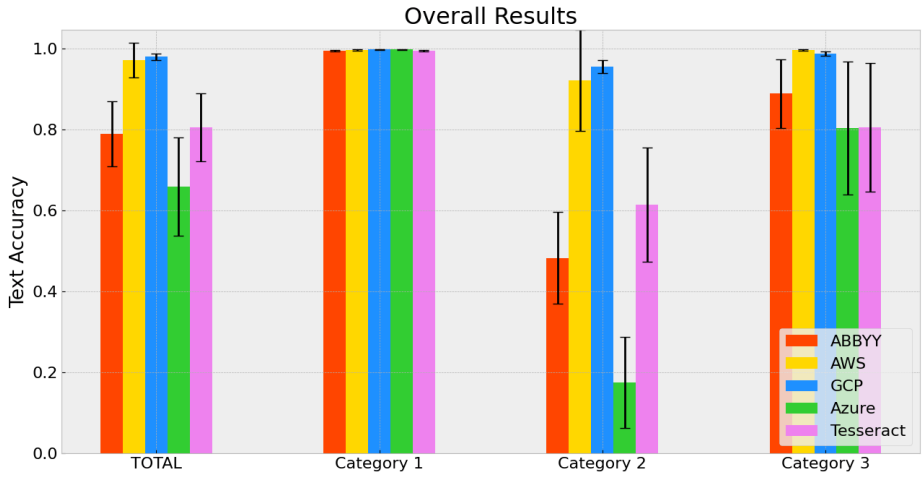
Sample outputs:

Results and discussions:

The post processing step in OCR was able to reduce the character and word error rates by more than 20%, showing its ability to remove burst-like noise that greatly disturbs the image partition. Moreover, added to the multiple hypotheses framework, the whole system yielded approximately 86% character recognition rate and a more than 90% word recognition rate on our database, which constitutes a reduction of more than 70% of error rate.

The detection algorithm is usually not affected by text color, font and size even the type of language. However, it will misclassify some texture regions as text and the detected boundaries of some text lines are not very accurate.

In our given input images some of the characters are not classified correctly because of some similarities with other characters like greater than (>) symbol is consider more similar to the character



Conclusion and future scope:

This paper presents a general scheme for extracting and recognizing embedded text of any gray-scale value in images and videos. The method is split into two main parts: the detection of text lines, followed by the recognition of text in these lines.

Applying machine learning methods for text detection encounters difficulties due to character size and gray-scale variations and heavy computation cost. To overcome these problem, we proposed a two-step localization/verification scheme. The first step aims at quickly locating candidate text lines, enabling the normalization of characters into a unique size.

References:

1. An Efficient Method for Text Detection and Recognition by Dr. M Meena kumari ,Dr. T. Mohanasundaram and R Suresh Kumar 2015
2. Image processing and find stroke width in natural images Amritha S Nadarajan&; Thamizharasi A (2018), proposes an innovative algorithm to find value of stroke width in natural images
3. B. Manjunath, W. Ma, Texture features for browsing and retrieval of image data, IEEE Trans. Pattern Anal. Mach. Intell.
4. Text detection and recognition in images and video frames by Jean-marc Odobez