A PROJECT REPORT

on

Business Card Reader

Submitted to

KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN
INFORMATION TECHNOLOGY

BY

KSHITIJ ARORA	1706047
SHUBHAM SHARMA	1706086
AMAN BHASKAR	1706107
AMRIT RAJ	1706109
KRISHNAKANT YADAV	1706135

UNDER THE GUIDANCE OF
PROF. SANKALP NAYAK

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024
May 2020

KIIT Deemed to be University

School of Computer Engineering

Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

Handwritten Character Recognition(HCR) Using Neural Network

submitted by

KSHITIJ ARORA	1706047
SHUBHAM SHARMA	1706086
AMAN BHASKAR	1706107
AMRIT RAJ	1706109
KRISHNAKANT YADAV	1706135

is a record of bonafide work carried out by them, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2019-2020, under our guidance.

Date: 07/06/2020

(Prof. SANKALP NAYAK)

Project Guide

Acknowledgements

We are profoundly grateful to Prof. SANKALP NAYAK for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion. ....

KSHITIJ ARORA
SHUBHAM SHARMA
AMRIT RAJ
KRISHNAKANT YADAV
AMAN BHASKAR

ABSTRACT

In this Digital age everything is shifting to digital format and as some experts say "data is new oil" the whole civilization is shifting to digital platforms to hold their data.
In this Digital age our communication is primarily digital and all the contacts have shifted to either email's or mobile contacts so as in the early days people use to carry business cards of individuals they met so that they can contact them in the future, it has become hard for us to manage all these cards physically.
A business card usually contains the name , contact, email address and the physical address of that individual.
All our contacts are stored on the internet or our mobile devices and it gets really hard to access our business cards and input the details manually every time and this problem escalates when the quantity of the cards increases.

The main aim of this project is to design a system that allows the user to scan their business cards and by using the Artificial Neural Network approach store the main contact information on their mobile devices. Neural computing Is comparatively new field, and design components are therefore less well specified than those of other architectures. Neural computers implement data parallelism. Neural computer are operated in way which is completely different from the operation of normal computers. Neural computer are trained (not Programmed) so that given a certain starting state (data input); they either classify the input data into one of the number of classes or cause the original data to evolve in such a way that a certain desirable property is optimized.
The system scans the data and stores in a database that can be further accesed and modified for future refrenses.

Keywords: Text Recognition,Handwritten Character,Pattern Recognition ,Feature Extraction,Neural Network.
.

Contents

Chapter 1

Introduction

1.1 BASIC

Text recognition is the ability of to scan a photograph and recognise the text from the image and output them.

Neural computing Is comparatively new field, and design components are therefore less well specified than those of other architectures.Neural computers implement data parallelism.Neural computer are operated in way which is completely different from the operation of normal computers.Neural computer are trained (not Programmed) so that given a certain starting state(data input); they either classify the input data into one of the number of classes or cause the original data to evolve in such a way that a certain desirable property is optimized.

1.1.1 Project Definition

This application is useful for recognizing all character(English) given as in input image.Once in put image of character is given to proposed system, then it will recognize input character which is given in image.Recognition and classification of characters are done by Neural Network.The main aim of this project is to effectively recognize a particular character of type format using the Artificial Neural Network approach.

1.2 Relevant Theory

1.2.1 Benefits of Character Recognition:

1. The idea of Neural Network in character recognition will brings us the reading of various combined style of writing a character.

2. In forensic application character recognition will be an effective method for

evidence collection.

3. .It will also help to reduce noise from the original character.

4. Our method develop accuracy in recognizing character in divert font and size.
5. More set of sample invites more accuracy rate because of heavy training and testing session.

1.2.2 Implementation of OpenCV :

OpenCV (**Open** Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. In simple language it is library used for Image Processing. It is mainly used to **do** all the operation related to Images.
OpenCV is used for image processing where it first recognizes the edges and then it saves the image into a black and white format for the next tecnology to use it which is pyteserract .

1.2.3 Pyteserract

An **Python-tesseract** is an optical character recognition (OCR) tool for python. That is, it will recognize and "read" the text embedded in images. ... Additionally, if used as a script, **Python-tesseract** will print the recognized text instead of writing it to a file.

This is used to scan the image that is saved by the OpenCV and then it uses the libraries and functions that are inbilt in it to recognise the text in the image .

The images are then stored in the master database which then uses the Neural networks to decode the details in it and provide the user with the details.
As of today, Tesseract can detect over 100 languages and can process even right-to-left text such as Arabic or Hebrew! No wonder it is used by Google for text detection on mobile devices, in videos, and in Gmail's image spam detection algorithm.

From version 4 onwards, Google has given a significant boost to this OCR engine. Tesseract 4.0 has added a new OCR engine that uses a neural network system based on LSTM (Long Short-term Memory), one of the most effective solutions for sequence prediction problems. Although its previous OCR engine using pattern matching is still available as legacy code.

1.2.4 What is Neural Network

An Artificial Neural Network (ANN) is an information-processing paradigm that is in- spired by the way biological nervous systems, such as the brain, process information.The key element of this paradigm is the novel structure of the information processing system. It is composed of large no. of highly interconnected processing element (neurons) work- ing in union to solve specific problems. ANN‘s like peopling, learning by example. An ANN is configured for a specific application. such as pattern recognition or data classification, through a learning process .Learning in a Biological system involves adjustments to the synaptic connections that exist between the neuron.

➢ Why use Neural Network

Neural network with their remarkable ability to derive meaning from complicated or im- precise data can be use to extract pattern and detect trend that are too complex to be noticed by either human or other computer techniques. A trained neural network can be thought of as an “expert” in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer “what if” questions. Other Advantages Include:

- Adaptive Learning: An ability to learn how to do tasks based on the data given for training or initial experience.

- Self-Organization:An ANN can create its own organization or representation of the information it receives during learning time.
- Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
- Fault Tolerance Via Redundant Information coding: partial destruction of network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

Chapter 2

Literature Survey

Text Capture and Optical Character Recognition Market: Global Growth Manufacturers, Regions, Product Types, Major Application Analysis & Forecast to 2025

The research report on Text Capture and Optical Character Recognition market provides with a granular evaluation of the business space and contains information regarding the market tendencies such as the prevailing remuneration, revenue estimations, market valuation and market size during the estimated timeframe.

An overview of the performance assessment of the Text Capture and Optical Character Recognition market is mentioned in the report. The document also comprises of insights pertaining to the major market trends and its predicted growth rate. Additional details such as growth avenues as well as hindering factors for this industry landscape are enlisted.

Recognition of Hindi Characters using Backpropagation Neural Network

Automatic recognition of handwritten characters is a difficult task because characters are written in various curved and cursive ways, so they could be of different sizes, orientation, thickness, format and dimension. An offline Hindi character recognition system using neural network is presented in this paper. Neural networks are good at recognizing h characters as these networks are insensitive to the missing data. The paper proposes the approach to recognize Hindi characters in four stages 1) Scanning, 2) Prepossessing, 3) Feature Extraction and, 4) Recognition. Preprocessing includes noise reduction, binarization, normalization and thinning. Feature extraction includes extracting some useful information out of the thinned image in the form of a feature vector. The feature vector comprises of pixels values of normalized character image. A Back propagation neural net- work is used for classification. Experimental result shows that this approach provides better results as compared to other techniques in terms of recognition accuracy, training time and classification time. The average accuracy of recognition of the system is 93 %.[2]

Contemporary Deep Learning Model

EAST, or Efficient and Accurate Scene Text Detector, is a deep learning model for detecting text from natural scene images. It is pretty fast and accurate as it is able to detect 720p images at 13.2fps with an F-score of 0.7820.

The model consists of a Fully Convolutional Network and a Non-maximum suppression stage to predict a word or text lines. The model, however, does not include some intermediary steps like candidate proposal, text region formation, and word partition that were involved in other previous models, which allows for an optimized model.

You can have a look at the image below provided by the authors in their paper comparing the EAST model with other previous models:

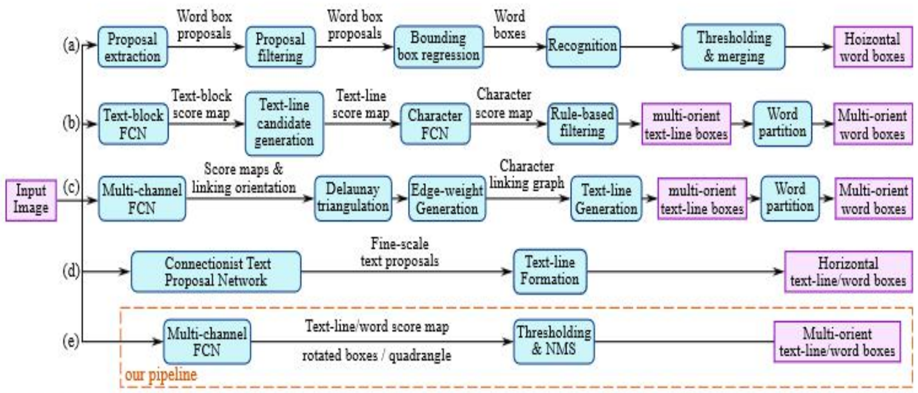


Figure 2. Comparison of pipelines of several recent works on scene text detection: (a) Horizontal word detection and recognition pipeline proposed by Jaderberg *et al.* [12]; (b) Multi-orient text detection pipeline proposed by Zhang *et al.* [48]; (c) Multi-orient text detection pipeline proposed by Yao *et al.* [41]; (d) Horizontal text detection using CTPN, proposed by Tian *et al.* [34]; (e) Our pipeline, which eliminates most intermediate steps, consists of only two stages and is much simpler than previous solutions.

EAST has a U-shape network. The first part of the network consists of convolutional layers trained on the ImageNet dataset. The next part is the feature merging branch which concatenates the current feature map with the unpooled feature map from the previous stage.

This is followed by convolutional layers to reduce computation and produce output feature maps. Finally, using a convolutional layer, the output is a score map showing the presence of text and a geometry map which is either a rotated box or a quadrangle that covers the text. This can be visually understood from the image of the architecture that was included in the research paper:

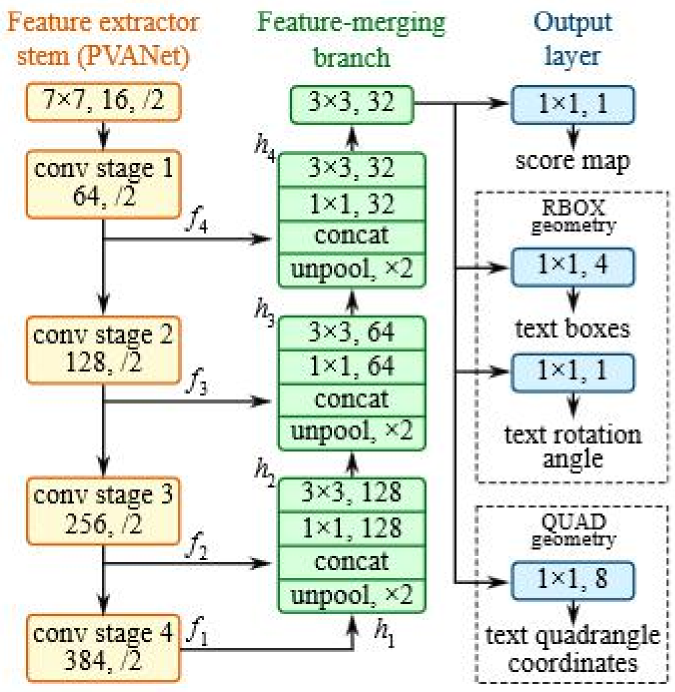
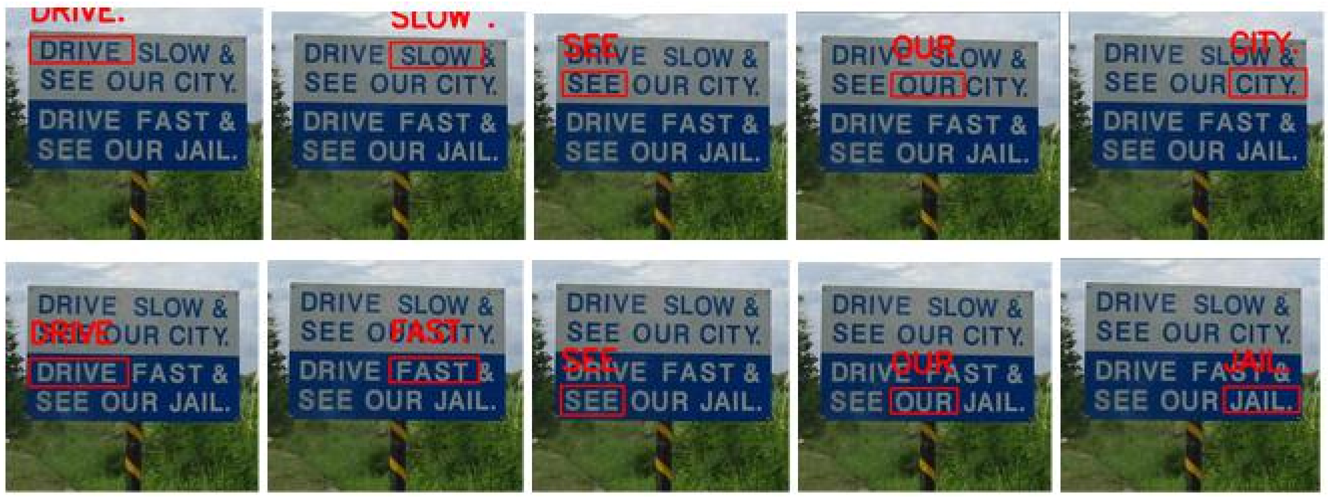


Figure 3. Structure of our text detection FCN.

OpenCV has included the EAST text detector model in version 3.4 onwards. This makes it super convenient to implement your own text detector. The resulting localized text boxes can be passed through Tesseract OCR to extract the text and you will have a complete end-to-end model for OCR.



1.1.2 Recognition for English Letters: A Review

Character recognition is one of the most interesting and challenging research areas in the field of Image processing. English character recognition has been extensively studied in the last half century. Nowadays different methodologies are in widespread use for character recognition. Document verification, digital library, reading bank deposit slips, reading postal addresses, extracting information from cheques, data entry, applications for credit cards, health insurance, loans, tax forms etc. are application areas of digital document processing. This paper gives an overview of research work carried out for recognition of character recognition.. In Hand written text there is no constraint on the writing style. Hand written letters are difficult to recognize due to diverse human handwriting style, variation in angle, size and shape of letters. Various approaches of hand written character recognition are discussed here along with their performance

Chapter 3

Software Requirements Specification

3.1Requirement Specification

3.1.1 Functional Requirements

1 The functional requirements for a system describe what system do.

2 The developed system should recognize English character present in the image.

3 System shall show the error message to the user when given input is not in the required format.

4 System must provide the quality of service to user.

5 System must provide accuracy for character recognition.

3.1.2 Performance

Response time for a transaction : average ~ 5 seconds , maximum ~ 10 seconds Throughput : 100 words/second Capacity : Limited Milli Piyango Web Site's simultaneous connection capacity Degradation mode : Only Push Notification when there is no Internet connection Resource utilization : Memory: %5 Disk: %1 Communications: %0.1

These are the requirements clearly stated by the customer hence requirement must be present for customer satisfaction.

N1 : Application should have graphical user interface.

N2 : Input of characters with various font size and styles should recognize.

N3 : Database should identify computer based English character by comparison .

N4 : Application should be able for matching the stored patterns on input handwritten character.

N5 : Minimum 10*50 (characters * patterns) should be available for each character.

3.1.3 Expected Requirements

These requirements are implicit type of requirements.These requirements are not clearly stated by customer but even though customer expects them.

Exp1 : Instead of only one character application should take set of characters or text.

Exp2 Application should be user friendly.

Exp3 : By using Neural Network bringing more accuracy in character recognition process.

Exp4 : Application also recognize English numerals.

3.2.1 Nonfunctional requirements

As the name suggest these are the requirements that are not directly interacted with specific functions delivered by the system.

Functionality: This software will deliver on the functional requirements.

Availability: This system will retrieve the handwritten text regions only if the image contains written text in it.

Flexibility: It provides the users to load the image easily.

Learn ability: The software is very easy to use and reduces the learning work.

Chapter 4

Requirement Analysis

Usability Since the users are assumed to have the common knowledge of using an easy smart phone application, there are no specific usability requirements. We guess it takes no more time to become productive on using this application than using an e-mail application.

With respect to the system under consideration the following issues are to be validated to ensure consistency of system.

4.1 System Requirements

•

4.1.1 Hardware Requirements

- Intel i3 Processor
- 128 MB RAM
- 10 GB Hard Disk.
- For the product to run, you shall need an Android device. To get accurate results, your device should have a high resolution camera. Also, since image processing is resource-hungry, your device should have modern RAM and CPU. Since neither the mobile application nor the web portal have any designated hardware, it does not have any direct hardware interfaces. The physical camera is managed by the camera interface in the smart phone and the hardware connection to the Internet is managed by the underlying operating system on the mobile phone and the web server.

4.1.2 Software Requirements

- Windows 7/8/8.1
- Language: Java(J2SE) JDK 1.7.
- As it has already been stated, product shall run on only Android devices, so it shall not need any specific software from the user's side other than it. The mobile application communicates with the web server in order to get lottery results. However, product shall depend on some libraries and tools to be developed. User interface module to be implemented shall interact with user and according to its input, shall activate image processing module. Image processing module shall be implemented with OpenCV libraries which prepares input for OCR operation. Tesseract OCR, which is Open Source free software, shall convert numbers on image to text accurately. Web module shall connect to Milli Piyango Web Site and shall gather related data. User interface module to be implemented shall display the results

Supportability

Data and operations in classes shall fit together. There shall not be extraneous dependencies between classes. Comments shall explain why things are done rather than what is done. Unit tests for every single class shall be implemented. Code repetition shall be avoided. Since primary development environment requires Java language, we shall adopt suggested name conventions by Sun Microsystems where a name in “CamelCase” is one composed of a number of words joined without spaces, with each word's initial letter in capitals. We shall use OpenCV, Android SDK,NDK, Tesseract OCR.

Chapter 5

System Design

5.1 Process Model

Process Model are processes of the same nature that are classified together into a model. Thus, a process model is a description of a process at the type level. Since the process model is at the type level, a process is an instantiation of it. The same process model is used repeatedly for the development of many applications and thus, has many instantiations. One possible use of a process model is to prescribe how things must/should/could is done in contrast to the process itself which is really what happens. A process model is roughly anticipation of what the process will look like. What the process shall be determined during actual system development.

The goal of a process model is to be:

- **Descriptive**
  1. Track what actually happens during a process.
  2. Take the point of view of an external observer who looks at the way a process has been performed and determines the improvements that must be made to make it perform more effectively or efficiently.
- **Prescriptive**

  1.Define the desired processes and how they should/could/might be performed. 2.Establish rules, guidelines, and behavior patterns which, if followed, would lead to the desired process performance. They can range from strict enforcement to flexible guidance.
- **Explanatory**
  1. Provide explanations about the rationale of processes.
  2. Explore and evaluate the several possible courses of action based on rational arguments.
  3. Establish an explicit link between processes and the requirements that the model needs to fulfil.
  4. Pre-defines points at which data can be extracted for reporting purposes

5.1.1 Incremental Model

Incremental model is used as the process model in our system. Figure 3.1 shows the process model of the system. To save actual problems in an industry setting, Software Engi- neering must incorporate a development strategy that encompasses the process, method and the tool layers; this strategy is often referred as process model. A process model for Software Engineering is chosen base on the nature of the Project and its application. For our project, we have selected Incremental Model.

1.Using these models, a limited set of customer requirements are implemented quickly and are delivered to customer.

2.Modified and expanded requirements are implemented step by step.

3.It combines elements of linear Sequential Model with the iterative Philosophy of prototyping.

4.Each linear sequence produces a deliverable Increment of the

Software. 5.Each linear Sequence is divide into 4 parts:-

- Analysis
- Design
- Code
- Testing

1. **Analysis**
   It includes understanding of information domain, required functions, behavior, performance and interface. Requirements for the system and software are documented and reviewed with customer.
2. **Design**
   It is multiple processes that include four attributes of program data structure, software architecture, interface representation and procedural detail.
3. **Coding**
   Translation of design to machine code is done by this step.
4. **Testing**
   It focuses on Logical internals of Software and ensures that all statement is correct to uncover all hidden errors. For an incremental model, the first Increment is developed as a core model, which is used by the customer. Then as things are added after the first delivery, product gets and better.

5.1.2 Advantages of Incremental Model

1. Generates working Software quickly and early during the software life cycle.
2. More Flexible-less costly to change Scope and Requirements.
3. Easier to test and debug during a smaller iteration.

5.1.3 Why we use Incremental Model?

The main aim of using the model is the reason that we have to add more features in the existing modules to increase project reliability and usability. Using this model we can adapt to the changing requirements of the customer which helps in developing the project in relatively small amount of time. The next increment implements customer suggestions plus some additional requirements in the previous increment. The process is repeated until the project is completed.

Chapter 6

Project Planning

The prerequistics for the following project are :

1. Knowledge of Python and libraries

2. Knowledge of **OpenCV** : OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

3. Knowledge of **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.

4. **NLP**(Natural Language Processing) : Natural language processing is a branch of computer science that uses artificial intelligence to interpret and process natural language. These techniques are used to process text, such as text messages, emails, and Web pages, in order to understand, interpret, and/or provide a response to the text. It is a method of extracting meaning from text. NLP is used to understand the structure of language. The idea is that if you have a large corpus of text, you can train a computer to recognize patterns in that corpus. The goal is to extract meaning from text, and to do this, we must be able to understand what the text is trying to say.

The building of extractor with **openCV** and **TesseractOCR** can be done by following steps :

Step 1 : The first step is about processing the image and making it perfect for the ocr to recognise text. It has the

- We have started by detecting the edges of the document that we want to scan.
- Using these edges, we have find the outline representing the piece of document scanned.
- Now we have applied a perspective transform to obtain the **top-down** view of the document.

Step 2 : We have applied **NLP** and **regex** to identify only the email and phone number in the extracted text and save the output.

Step 3 : Then we have used **pytesseract** to extract the text from the scanned image.

Step 1 : Processing of image input

We have used a perspective transform to obtain a top-down, “birds eye view”(an elevated view of an object from above) of the image.
We have used perspective transform to obtain a top-down, “bird’s eye view” of the image.Now Open a file and named fpt.py which will include the code for perspective transform.

The necessary packages that we used are :
1. **Numpy** for numerical processing
2. **cv2** for OpenCV bindings

Then we have defined the order_points function that takes a single argument pts, which is a list of four points specifying (x, y) coordinates of each point of the document.
It is absolutely crucial that we have a consistent ordering of the points in the rectangle.
We have specified points in the top-left, top-right, bottom-right and bottom-left order.
Then we have initialised a list of coordinates that will be ordered such that the first entry in the list is the top-left, the second entry is the top-right, the third is the bottom-right, and the fourth is the bottom-left.After that we have allocated memory for the four ordered points.
Then, we have find the top-left point, which will have the x + y sum, and the bottom-right point, which will have the largest x + y sum.Now we have to find the top-right and bottom-left points by taking the difference (i.e. x – y) between the points using the np.diff function.The coordinates associated with the smallest difference will be the top-right points, whereas the coordinates with the largest difference will be the bottom-left points.
Finally, we return our ordered functions to the calling function.

Four_point_transform function :
Here we started by defining the four_points_transform function which requeires two arguments “image” and “pts”.
The image variable is the image we want to apply the perspective transform to and the pts list is the list of four points that contain the ROI of the image we want to transform.
Then we have determine the dimensions of our new wrapped image. We have first determined it’s width, where the width is the largest distance between the bottom-right and bottom-left X-coordinates or the top-right and top-left X-coordinates. Similarly we have determined the height of the image, where the height is the maximum distance between the top-right and bottom-right y-coordinates or the top-left and bottom-left y-coordinates.

Now that we have determined the dimensions of the image, we constructed the set of destination points to obtain a “birds eye view”,(i.e. top-down view) of the image, again specifying points in the top-left, top-right, bottom-right, and bottom-left order.

We have define 4 points representing our “top-down” view of the image. The first entry in the list is (0, 0) indicating the top-left corner. The second entry is (maxWidth – 1, 0) which corresponds to the top-right corner. Then we have defined (maxWidth – 1, maxHeight – 1) which is the bottom-right corner. Finally, we have defined (0, maxHeight – 1) which is the bottom-left corner.

To obtain the top-down, “bird’s eye view” of the image we have used the cv2.getPerspectiveTransform function. This function requires two arguments, rect , which is the list of 4 ROI points in the original image, and dst , which is our list of transformed points. The cv2.getPerspectiveTransform function returns M , which is the actual transformation matrix we want.

Then after getting the transformation matrix we then apply the transformation matrix using the cv2.warpPerspective function. Then we pass in the image , our transform matrix M , along with the width and height of our output image.

The output of cv2.warpPerspective is our warped image, which is our top-down view.

So whenever we need to perform a 4 point perspective transform, we will be using fpt.py module.

Building of Card Scanner :

The necessary packages that we used are :

1. four_point_transform
2. Imutils (function for resizing, rotating and cropping images)
3. Threshold_local function from scikit-image function (which help to obtain 'black and white' feel to scanned image)
4. NumPy (for numerical processing)
5. cv2 (for OpenCV bindings)

Step 1 : Edge detection, finding outline and performing perspective transform

First we load our image on disk and in order to speed up the image processing and as well as to make our edge detection step more accurate, we have resized our scanned image to have a height of 500 pixels.We have also taken care to keep track of the ratio of the original height of the image to the new height which this will allow us to perform the scan on the original image rather than the resized image.And then, we converted the image from RGB to grayscale, then at last perform Canny edge detection.

Then we start off by finding the contours in our edged image.
Then we loop over the contours and then approximate the number of points. If the approximated contour has four points, then we assume that we have found the document in the image.
After that display the contours of the document we want to scan.Now we apply perspective transform by taking the four points representing the outline of the document to obtain a top-down, "birds-eye view" of the image.Then we start by performing the warping transformation and all the heavy lifting is handled by the four_point_transform function.
Here as you can see, we have passed two arguments into four_point_transform function: first is the original image we are loading off the disk(not the resized one), and the second argument is the contour representing the document which is multiplied by the resized ratio.

We have multiplied contour by the resized ratio because we performed edge detection and found contours on the resized image of height=500 pixels.
If we want to perform scan on the original image, not the resized image, then we need to multiply the contour points by the resized ratio.And at the end to obtain the black and white feel to the image, we then take the warped image, convert it to grayscale and apply adaptive thresholding.

Finally, we display our output and save one copy on disk locally.

Step 3 : Extraction of text from final scanned image using Tesseract OCR

Before using Tesseract OCR library in system, it needs to be installed first.

Python-tesseract is an optical character recognition (OCR) tool for python. That is, it will recognize and “read” the text embedded in images.
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.

We first started by importing our tesseract library used for extracting text from our scanned image and also importing PIL package which we will be using for opening, manipulating, and saving many different image file formats.
Then we added tesseract to our path and use image_to_string method to return the result of a Tesseract OCR run on the image to a string and we use the English language to convert to text.

Step 2 : NLP and regex to identify contact details

NLP : We start by importing library nltk. Then we read the file that was saved after ocr. We tokenise all the characters and add tags to each of it according to the nltk named entity recognition. Next step is chunking . After all this the value of proper nouns are choosen and is saved in a list. Then anew file is opened and it is used to save those values

Regex : We start by importing the re library. A regular expression is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern.

“[a-z0-9\.\-+_]+@[a-z0-9\.\-+_]+\.[a-z]+” is the regular expression used to find emails in any given string pattern. We are using findall method to find emails in the tesseract run output.

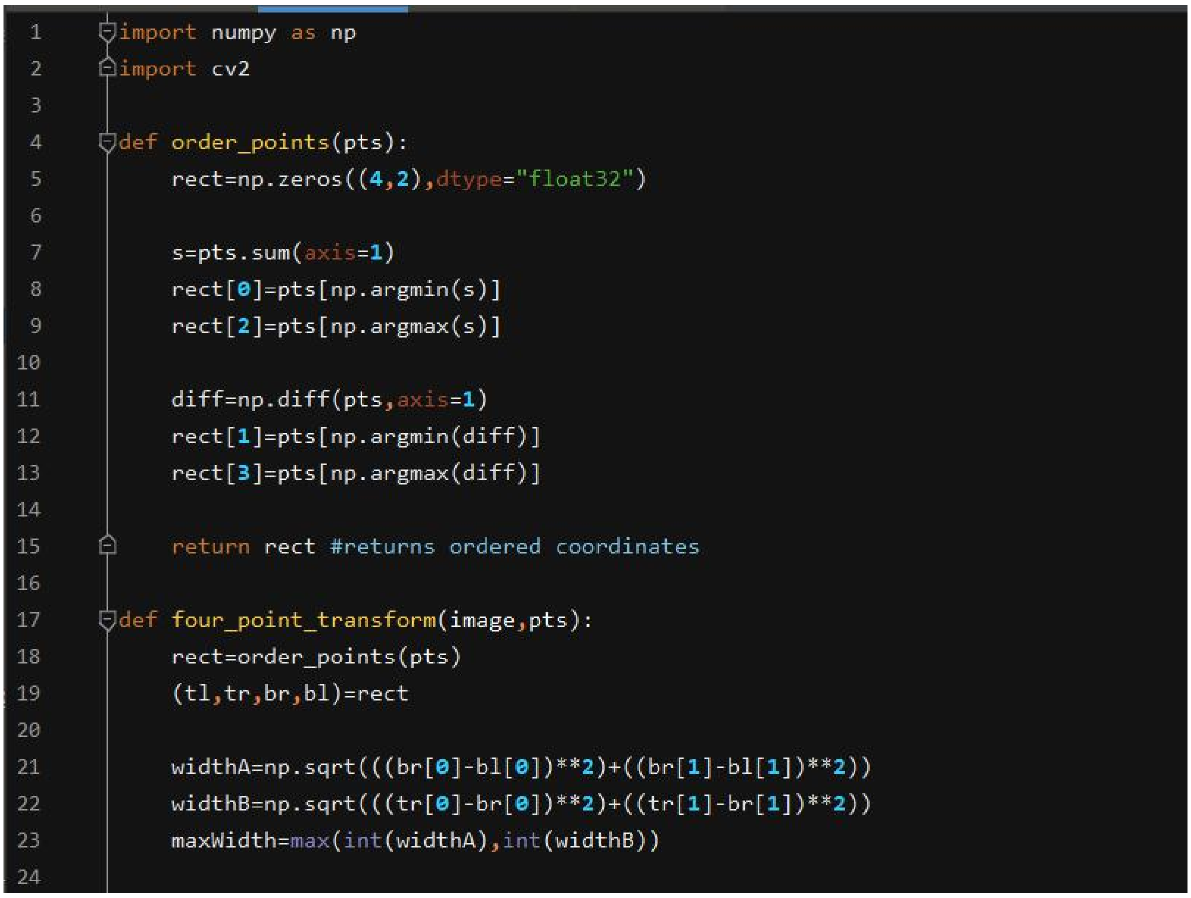
‘[\+]?[1−9][0−9.\-\(]?[1−9][0−9.\-\(]{8,}[0-9]’ is the regular expression used to find numbers:
*The matched string may start with + or (symbol
*It has to be followed by a number between 1-9
*It has to end with a number between 0-9
*It may contain 0-9 (space) .-() in the middle.

It will return the list as an output. We then loop through all the extracted emails and phone number list and print the emails and phone number.

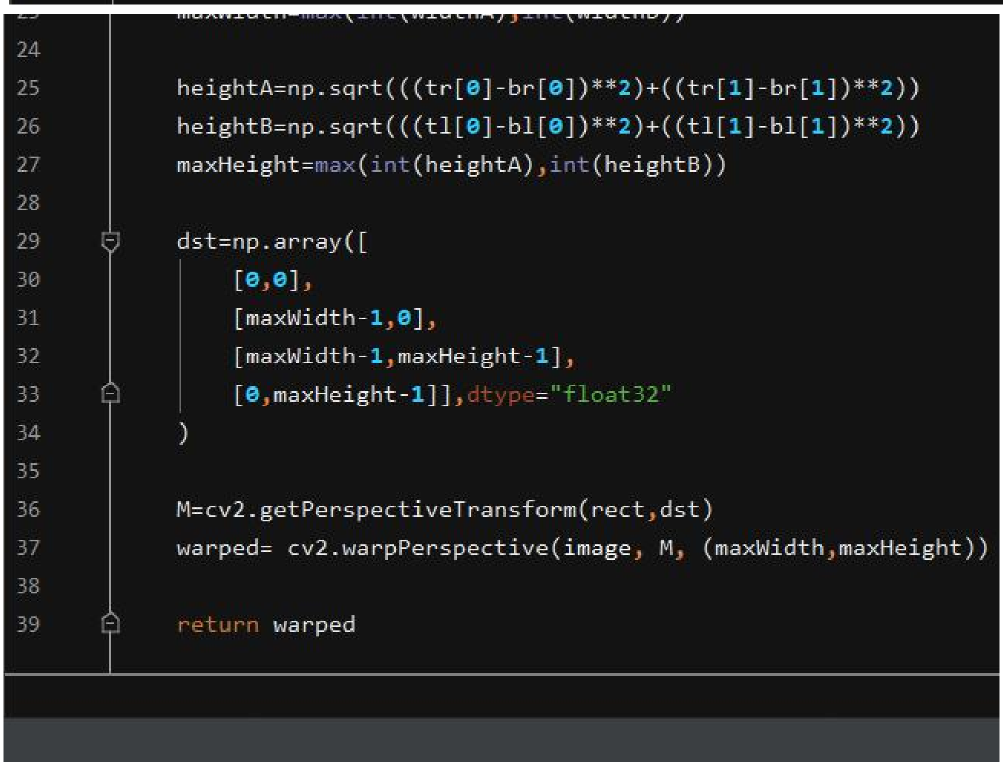
Chapter-7

Implementation

fpy.py

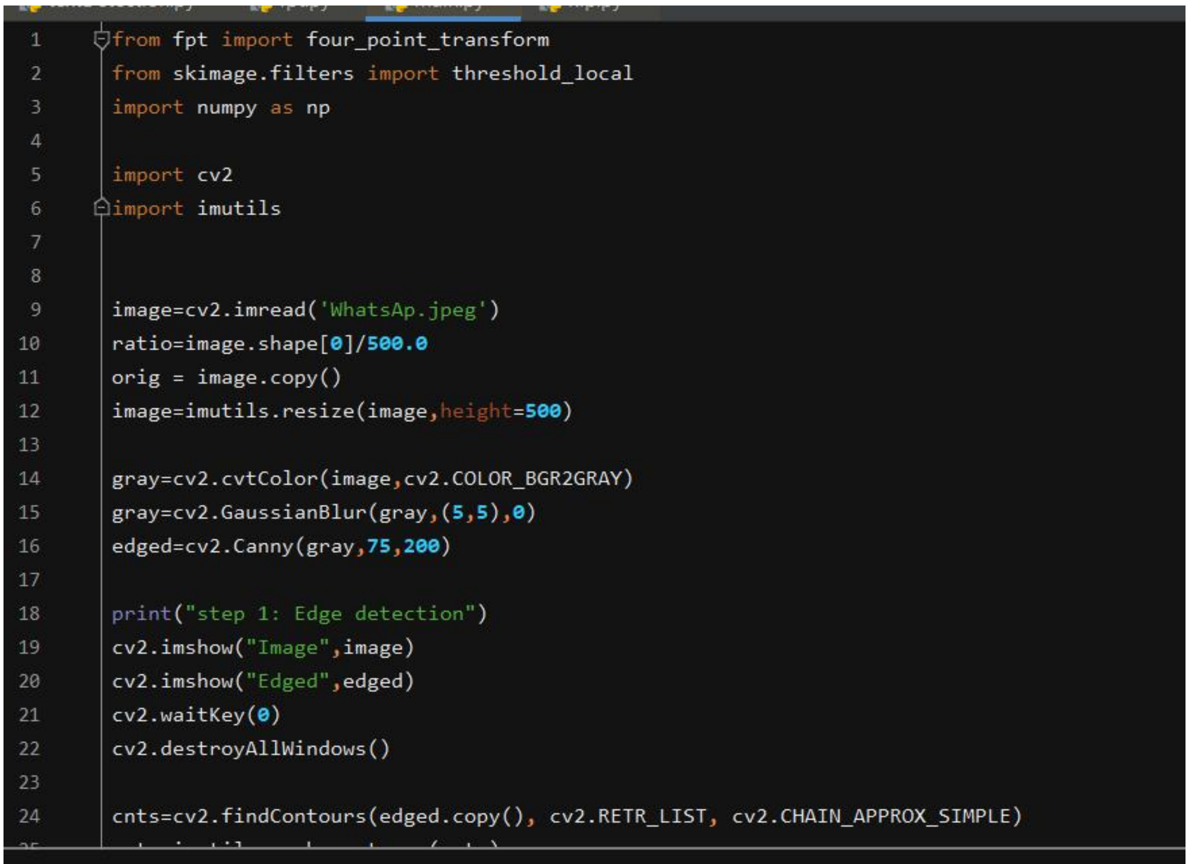


```python
import numpy as np
import cv2

def order_points(pts):
    rect=np.zeros((4,2),dtype="float32")

    s=pts.sum(axis=1)
    rect[0]=pts[np.argmin(s)]
    rect[2]=pts[np.argmax(s)]

    diff=np.diff(pts,axis=1)
    rect[1]=pts[np.argmin(diff)]
    rect[3]=pts[np.argmax(diff)]

    return rect #returns ordered coordinates

def four_point_transform(image,pts):
    rect=order_points(pts)
    (tl,tr,br,bl)=rect

    widthA=np.sqrt(((br[0]-bl[0])**2)+((br[1]-bl[1])**2))
    widthB=np.sqrt(((tr[0]-br[0])**2)+((tr[1]-br[1])**2))
    maxWidth=max(int(widthA),int(widthB))

```

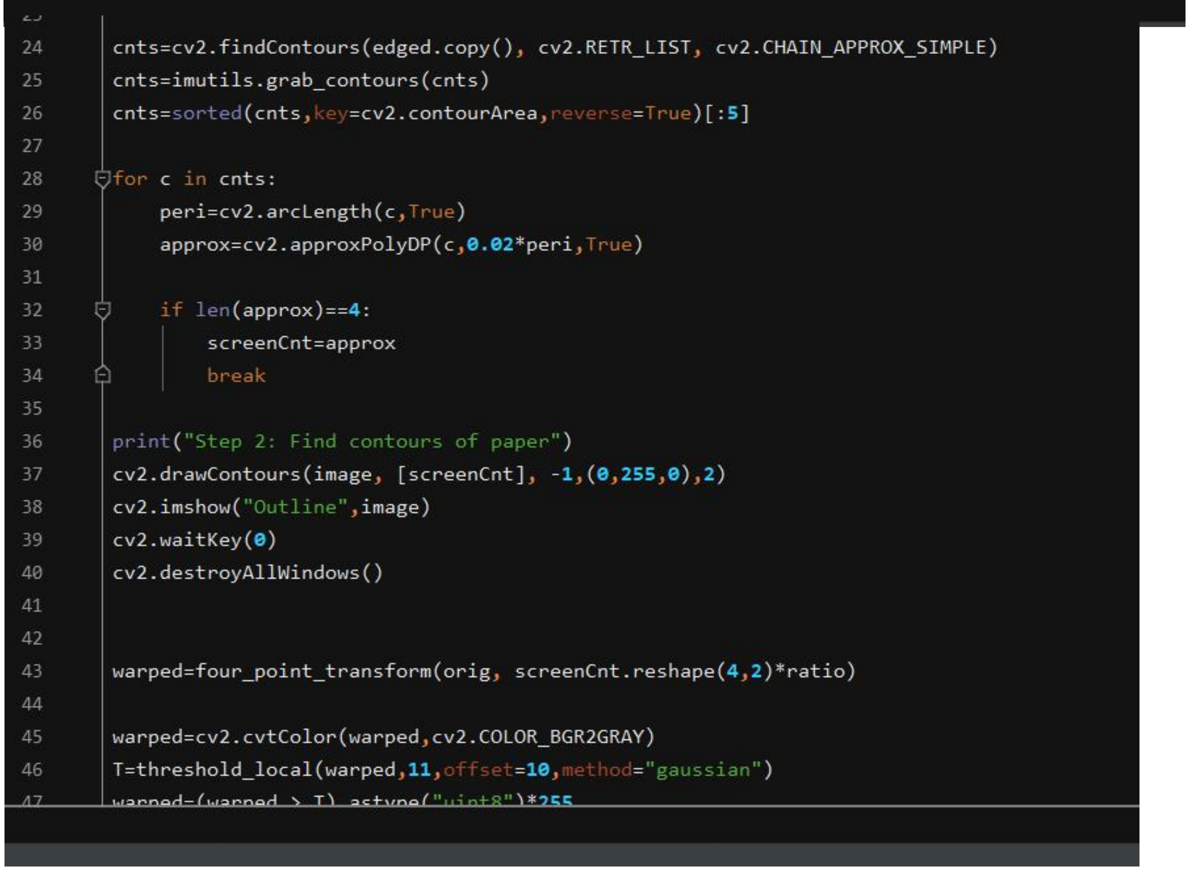


```python

    heightA=np.sqrt(((tr[0]-br[0])**2)+((tr[1]-br[1])**2))
    heightB=np.sqrt(((tl[0]-bl[0])**2)+((tl[1]-bl[1])**2))
    maxHeight=max(int(heightA),int(heightB))

    dst=np.array([
        [0,0],
        [maxWidth-1,0],
        [maxWidth-1,maxHeight-1],
        [0,maxHeight-1]],dtype="float32"
    )

    M=cv2.getPerspectiveTransform(rect,dst)
    warped= cv2.warpPerspective(image, M, (maxWidth,maxHeight))

    return warped
```

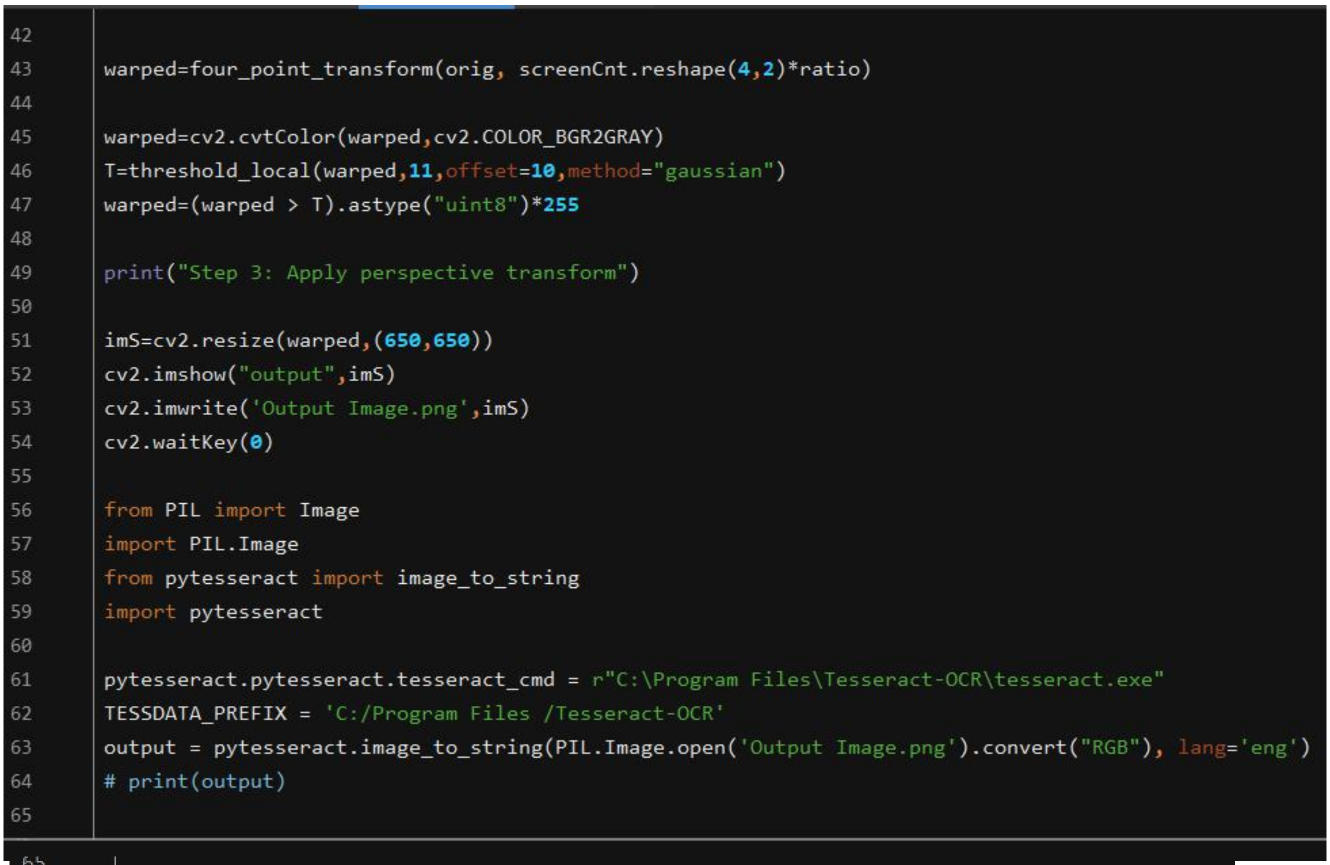
The main.py file



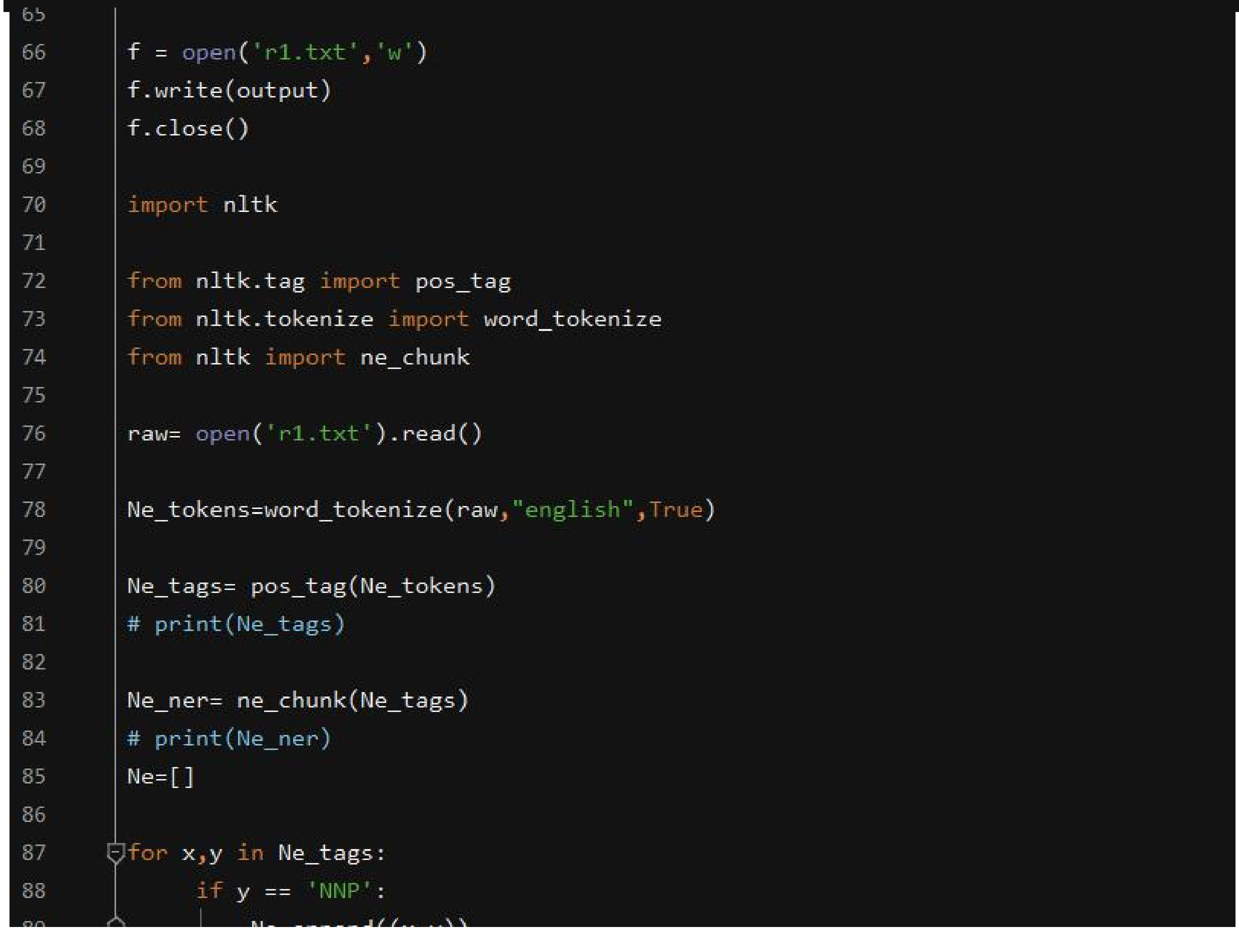
```python
from fpt import four_point_transform
from skimage.filters import threshold_local
import numpy as np

import cv2
import imutils


image=cv2.imread('WhatsAp.jpeg')
ratio=image.shape[0]/500.0
orig = image.copy()
image=imutils.resize(image,height=500)

gray=cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
gray=cv2.GaussianBlur(gray,(5,5),0)
edged=cv2.Canny(gray,75,200)

print("step 1: Edge detection")
cv2.imshow("Image",image)
cv2.imshow("Edged",edged)
cv2.waitKey(0)
cv2.destroyAllWindows()

cnts=cv2.findContours(edged.copy(), cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)
```



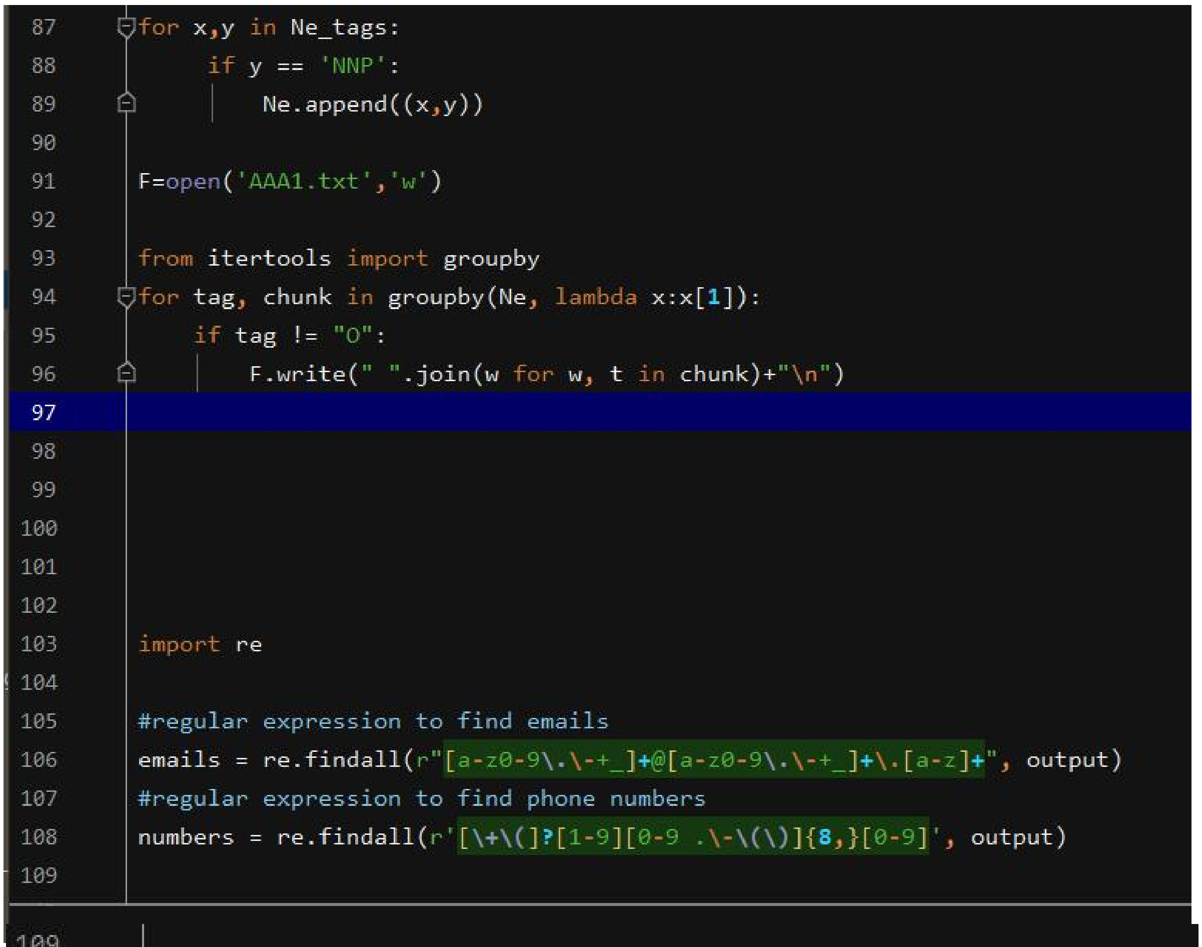
```python
cnts=cv2.findContours(edged.copy(), cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)
cnts=imutils.grab_contours(cnts)
cnts=sorted(cnts,key=cv2.contourArea,reverse=True)[:5]

for c in cnts:
    peri=cv2.arcLength(c,True)
    approx=cv2.approxPolyDP(c,0.02*peri,True)

    if len(approx)==4:
        screenCnt=approx
        break

print("Step 2: Find contours of paper")
cv2.drawContours(image, [screenCnt], -1,(0,255,0),2)
cv2.imshow("Outline",image)
cv2.waitKey(0)
cv2.destroyAllWindows()


warped=four_point_transform(orig, screenCnt.reshape(4,2)*ratio)

warped=cv2.cvtColor(warped,cv2.COLOR_BGR2GRAY)
T=threshold_local(warped,11,offset=10,method="gaussian")
warped=(warped > T).astype("uint8")*255
```



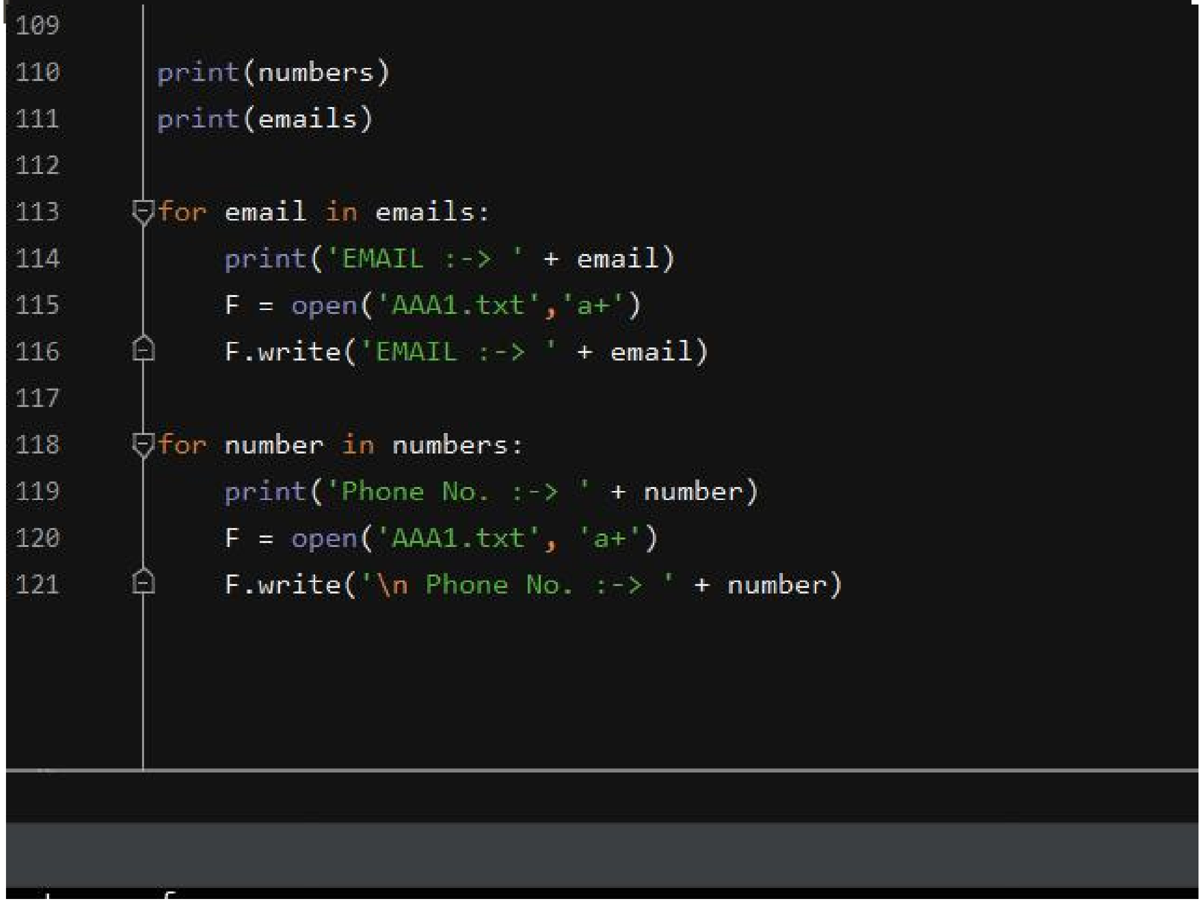
```
warped=four_point_transform(orig, screenCnt.reshape(4,2)*ratio)

warped=cv2.cvtColor(warped,cv2.COLOR_BGR2GRAY)
T=threshold_local(warped,11,offset=10,method="gaussian")
warped=(warped > T).astype("uint8")*255

print("Step 3: Apply perspective transform")

imS=cv2.resize(warped,(650,650))
cv2.imshow("output",imS)
cv2.imwrite('Output Image.png',imS)
cv2.waitKey(0)

from PIL import Image
import PIL.Image
from pytesseract import image_to_string
import pytesseract

pytesseract.pytesseract.tesseract_cmd = r"C:\Program Files\Tesseract-OCR\tesseract.exe"
TESSDATA_PREFIX = 'C:/Program Files /Tesseract-OCR'
output = pytesseract.image_to_string(PIL.Image.open('Output Image.png').convert("RGB"), lang='eng')
# print(output)

```



```

f = open('r1.txt','w')
f.write(output)
f.close()

import nltk

from nltk.tag import pos_tag
from nltk.tokenize import word_tokenize
from nltk import ne_chunk

raw= open('r1.txt').read()

Ne_tokens=word_tokenize(raw,"english",True)

Ne_tags= pos_tag(Ne_tokens)
# print(Ne_tags)

Ne_ner= ne_chunk(Ne_tags)
# print(Ne_ner)
Ne=[]

for x,y in Ne_tags:
     if y == 'NNP':
         Ne.append((x,y))
```



```python
for x,y in Ne_tags:
      if y == 'NNP':
        Ne.append((x,y))

F=open('AAA1.txt','w')

from itertools import groupby
for tag, chunk in groupby(Ne, lambda x:x[1]):
    if tag != "O":
        F.write(" ".join(w for w, t in chunk)+"\n")






import re

#regular expression to find emails
emails = re.findall(r"[a-z0-9\.\-+_]+@[a-z0-9\.\-+_]+\.[a-z]+", output)
#regular expression to find phone numbers
numbers = re.findall(r'[\+\(]?[1-9][0-9 .\-\(\)]{8,}[0-9]', output)

```



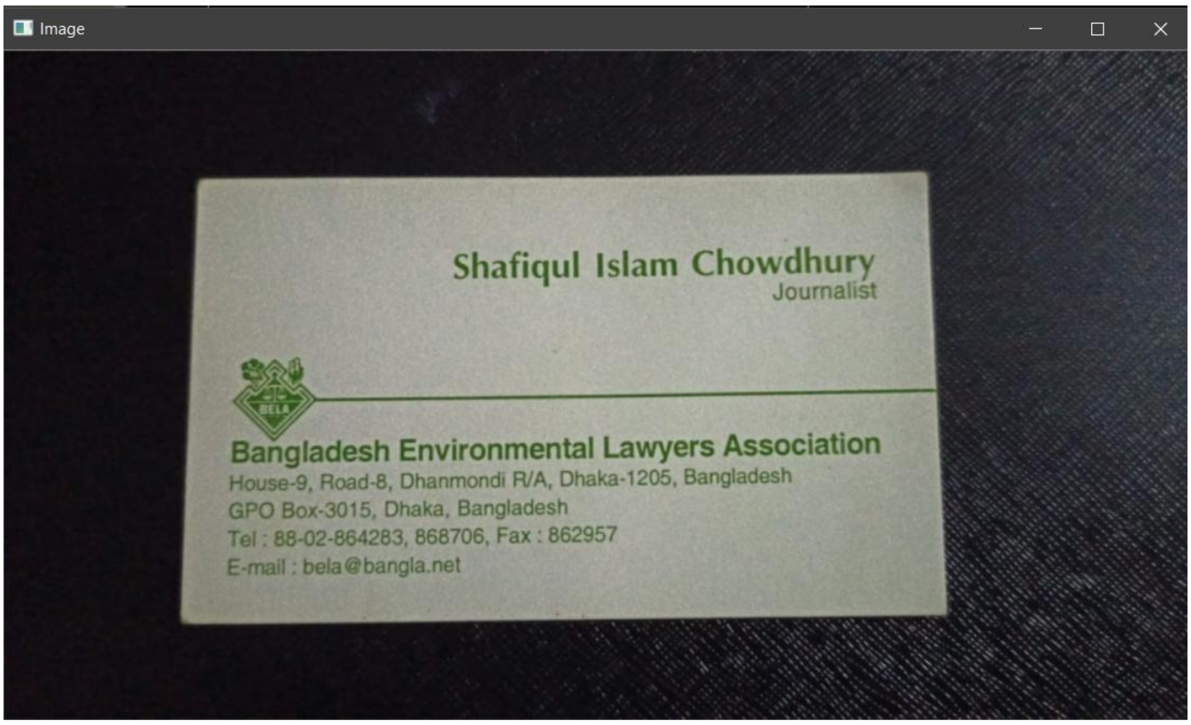
```python

print(numbers)
print(emails)

for email in emails:
    print('EMAIL :-> ' + email)
    F = open('AAA1.txt','a+')
    F.write('EMAIL :-> ' + email)

for number in numbers:
    print('Phone No. :-> ' + number)
    F = open('AAA1.txt', 'a+')
    F.write('\n Phone No. :-> ' + number)
```

Chapter-8

Results

Case 1

Input:

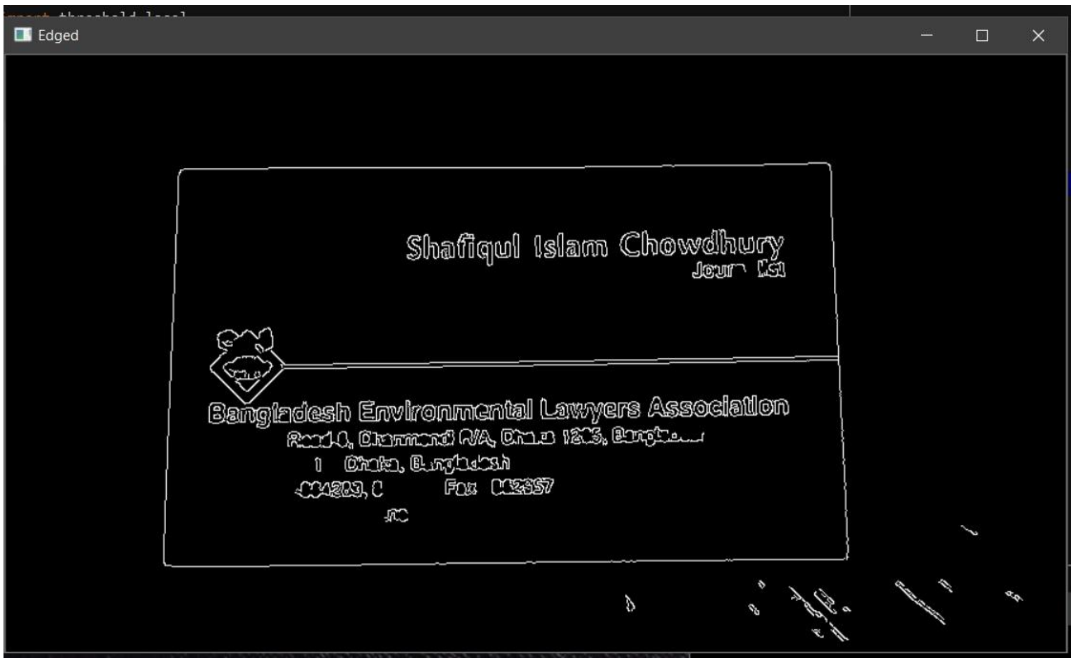


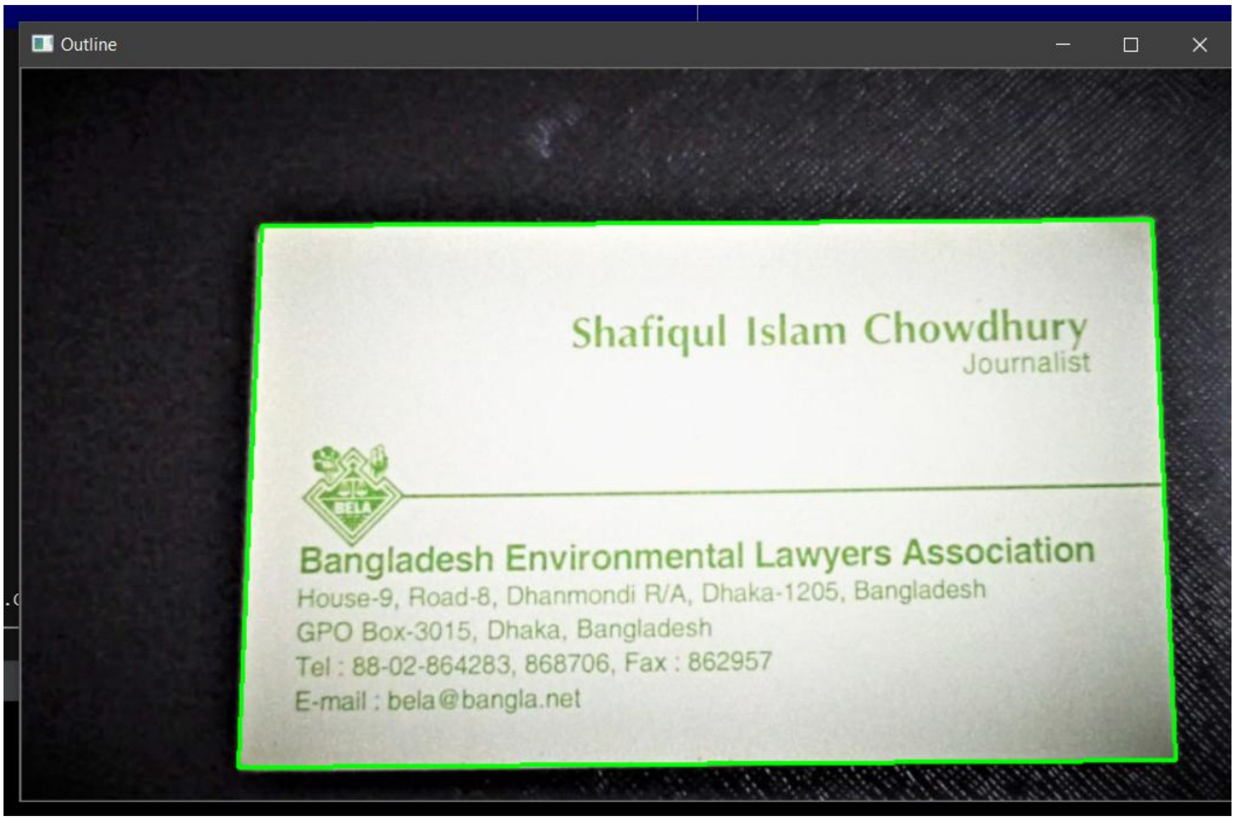
Shafiqul Islam Chowdhury
Journalist

BELA

Bangladesh Environmental Lawyers Association
House-9, Road-8, Dhanmondi R/A, Dhaka-1205, Bangladesh
GPO Box-3015, Dhaka, Bangladesh
Tel : 88-02-864283, 868706, Fax : 862957
E-mail : bela@bangla.net

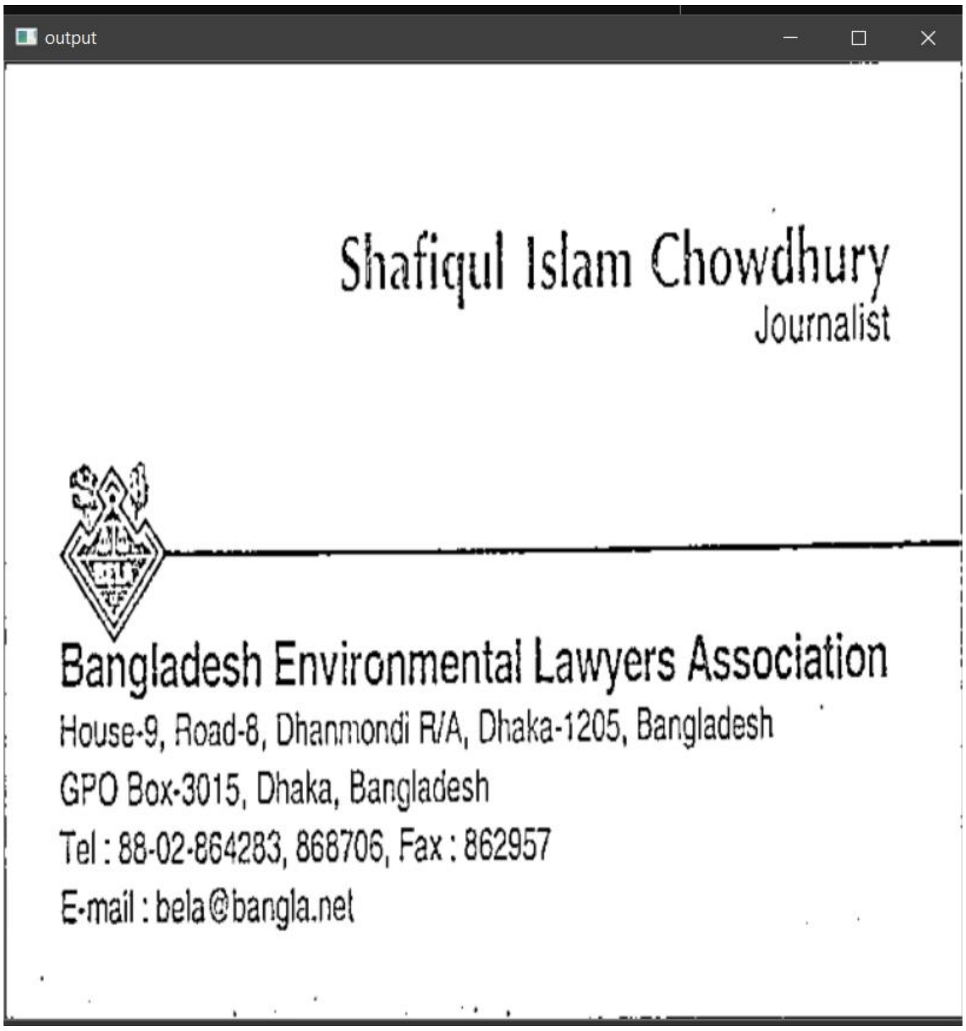
Outputs:





Shafiqul Islam Chowdhury
Journalist

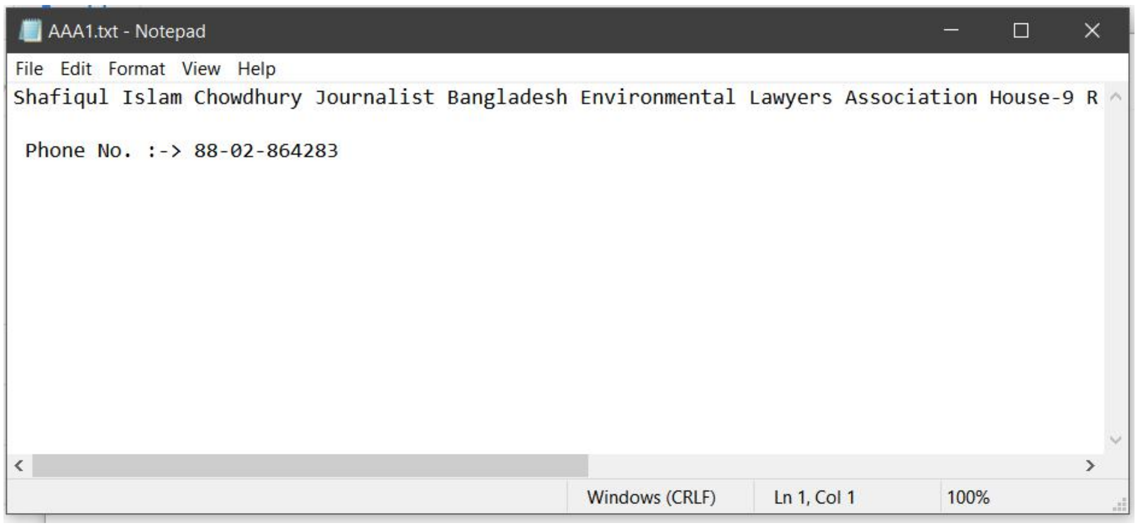
Bangladesh Environmental Lawyers Association
House-9, Road-8, Dhanmondi R/A, Dhaka-1205, Bangladesh
GPO Box-3015, Dhaka, Bangladesh
Tel : 88-02-864283, 868706, Fax : 862957
E-mail : bela@bangla.net



Shafiqul Islam Chowdhury
Journalist

Bangladesh Environmental Lawyers Association
House-9, Road-8, Dhanmondi R/A, Dhaka-1205, Bangladesh
GPO Box-3015, Dhaka, Bangladesh
Tel : 88-02-864283, 868706, Fax : 862957
E-mail : bela@bangla.net

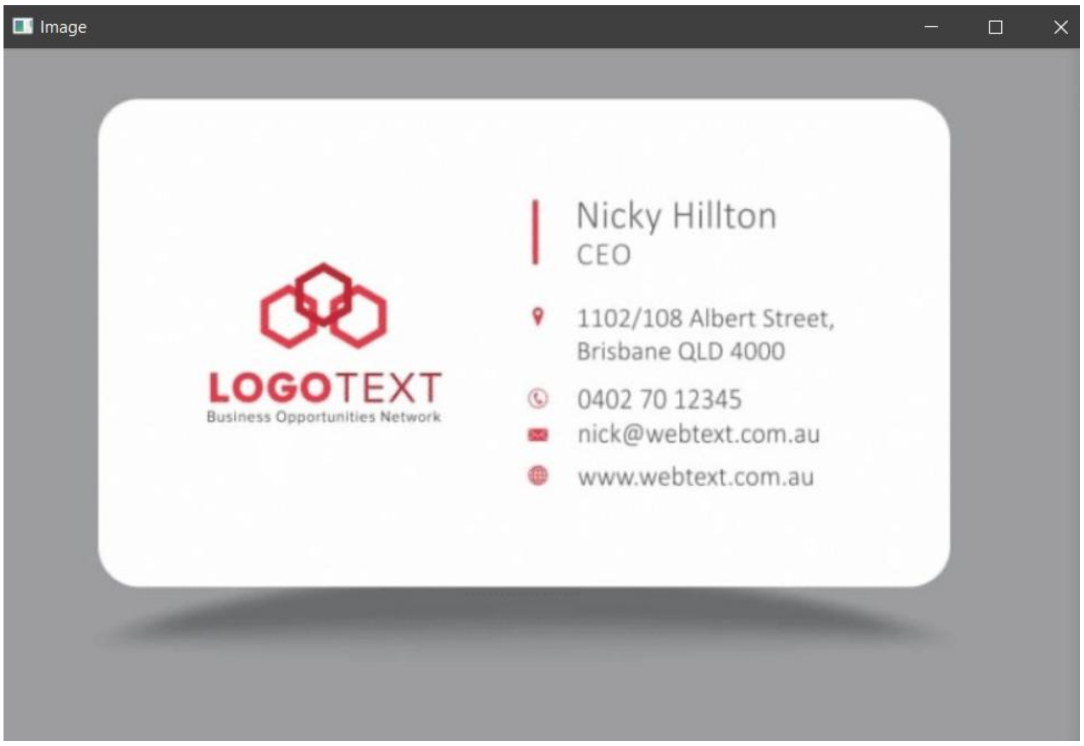
Final Output:


AAA1.txt - Notepad

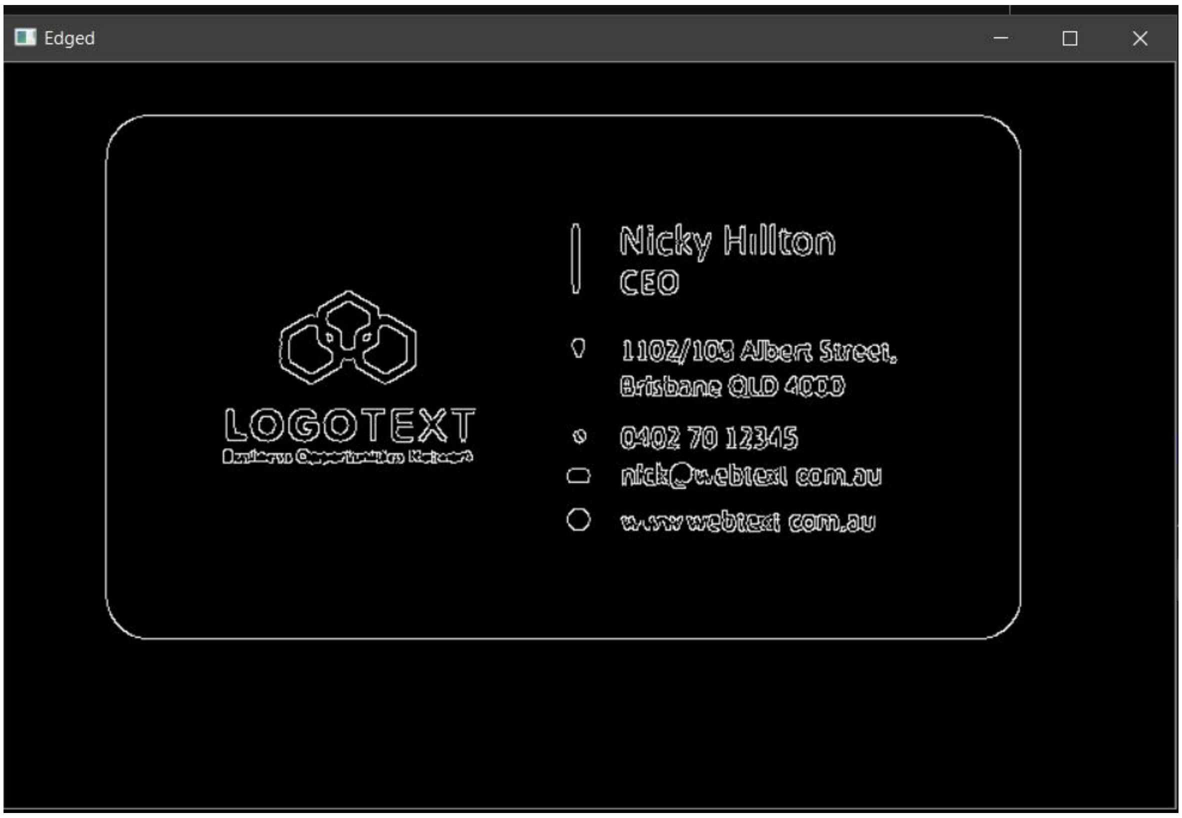
```
Shafiqul Islam Chowdhury Journalist Bangladesh Environmental Lawyers Association House-9 R

 Phone No. :-> 88-02-864283
```

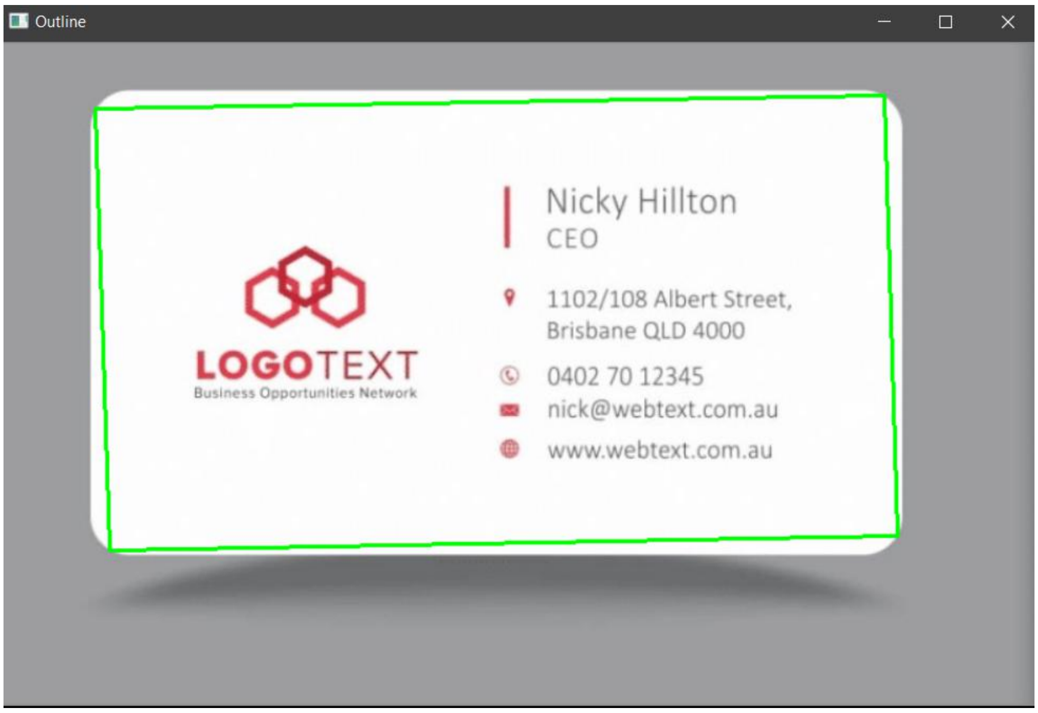
Case 2

Input:



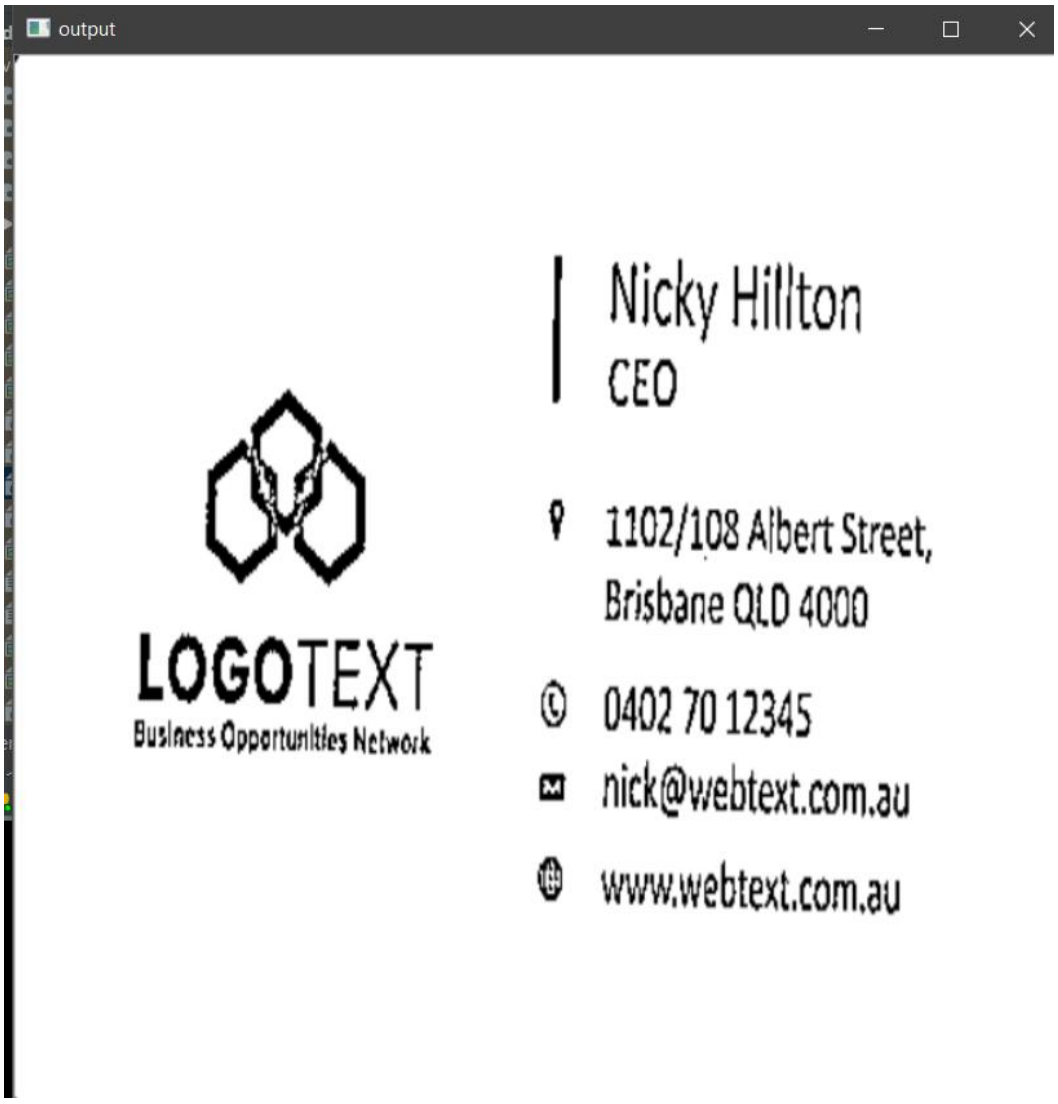
Outputs:





Outline

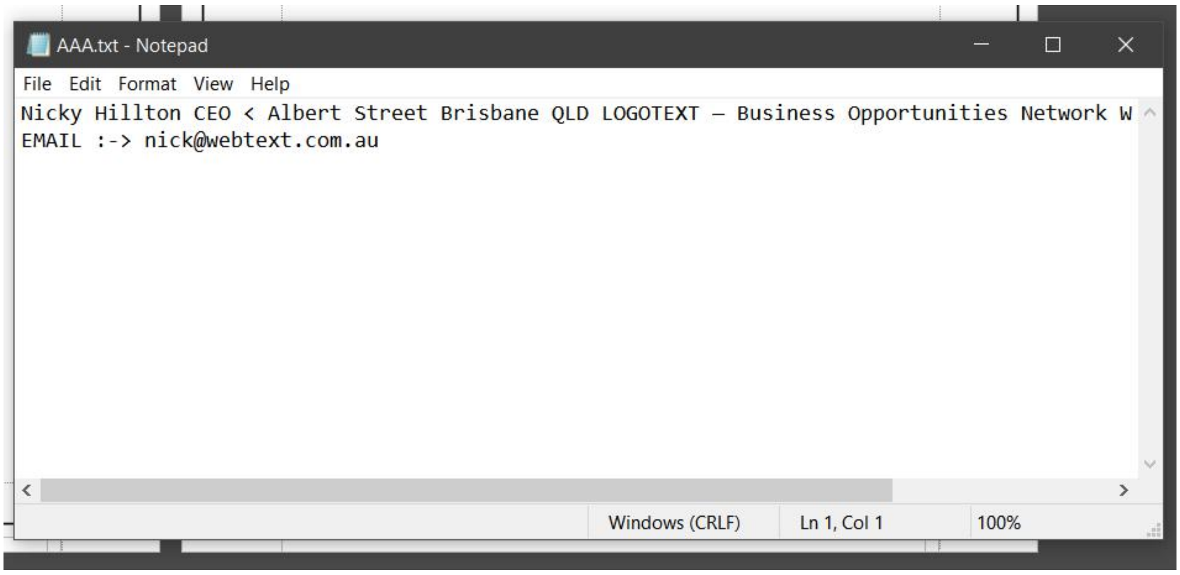
Nicky Hillton
CEO
1102/108 Albert Street,
Brisbane QLD 4000
0402 70 12345
nick@webtext.com.au
www.webtext.com.au
LOGOTEXT
Business Opportunities Network



output

Nicky Hillton
CEO
1102/108 Albert Street,
Brisbane QLD 4000
0402 70 12345
nick@webtext.com.au
www.webtext.com.au
LOGOTEXT
Business Opportunities Network

Final Output


AAA.txt - Notepad

```
Nicky Hillton CEO < Albert Street Brisbane QLD LOGOTEXT – Business Opportunities Network W
EMAIL :-> nick@webtext.com.au
```

Chapter-9

Future Scope

The Future of this Technology can be extended in vast fields from corporate meeting cards to vendors cards.
In the future you can also use it to Scan anything Physical text or document and make a digital version of it.
You can use to scan anything written to save it in a digital format so that if you loose it in future it is all saved in the digital format.
Prescriptions, Grocerery list,To do list Menues of restauraunts everything can be stored using this and it takes up very less memory compared to a photograph.

Chapter-10

Conclusion

The model was made it detected the info about the person from the business card and stored in the file. This system can be used by people to extract data from images.
This will help reduce our dependence on paper and help us in shifting to digital.

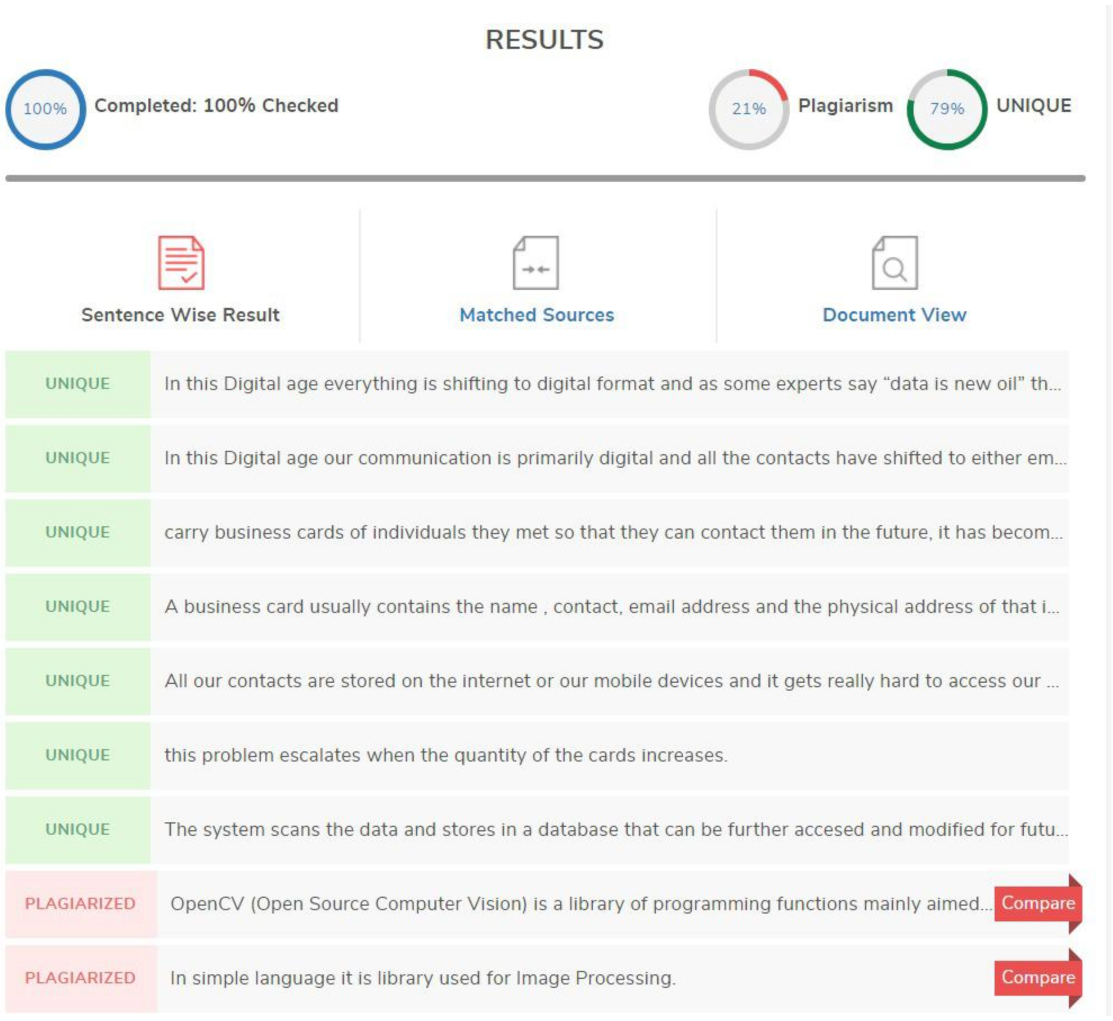
Chapter-11

References

- https://www.nltk.org/
- www.pyimagesearch.com › 2017/07/10
- https://github.com/opencv/opencv
- https://pythonprogramming.net/tokenizing-words-sentences-nltk-tutorial/
  https://www.tutorialspoint.com/python/python_reg_expressions.html

Chapter-12

Plagiarism Test



UNIQUE	In this Digital age everything is shifting to digital format and as some experts say "data is new oil" th...
UNIQUE	In this Digital age our communication is primarily digital and all the contacts have shifted to either em...
UNIQUE	carry business cards of individuals they met so that they can contact them in the future, it has becom...
UNIQUE	A business card usually contains the name , contact, email address and the physical address of that i...
UNIQUE	All our contacts are stored on the internet or our mobile devices and it gets really hard to access our ...
UNIQUE	this problem escalates when the quantity of the cards increases.
UNIQUE	The system scans the data and stores in a database that can be further accesed and modified for futu...
PLAGIARIZED	OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed...
PLAGIARIZED	In simple language it is library used for Image Processing.