Text Recognition Using CNN

Jaidev Chittoria   
*Information Technology Department*   
*National Institute of Technology Karnataka*Surathkal, India  
jaidev.chittoria02@gmail.com

Ayush Bhandari  
*Information Technology Department*   
*National Institute of Technology Karnataka*Surathkal, India  
ayushbhandari8909@gmail.com

Royce  
*Information Technology Department*   
*National Institute of Technology Karnataka*Surathkal, India

***Abstract*—*Text recognition is widely used technique used to recognise text from images or else the conversion of typed, written or printed text into machine encoding text.OCR is one way to get the recognized text from the image but as the image get complex and unstructured OCR didn’t produced good results In this Project Deep learning method ; Convolutional neural network is used which increases accuracy and precision of the recognised text so that it can give better results for natural images..***

***Keywords—Text Recognition, Deep Learning, Convolutional Neural Network, OCR, Natural images.***

# Introduction

The world is changing so fast, everything is being made digital from cash transactions to any documentation everything is being digitized and for converting these documents in digital form, text recognition plays an important role; recognizing text from an image is an important task that helps in storing text from any captured image real time image. It is widely used as a form of getting information from printed paper data records, it is a common method of digitizing printed texts so that they can be edited, searched, stored more compactly, displayed on screen, and used in machine processes such as cognitive computing, machine translation,(extracted) text-to-speech, key data and text mining. Getting text from a natural image is not an easy task, using simple methods, one has to type the text in a text editor by looking at the image which in turn waste so much of time and also by using other image recognition techniques there is a problem of lighting conditions,noise, image quality, and non-planar objects. Using text recognition from an image using deep learning methods like CNN, it performs better results on unstructured text and has low error rate.

# Literature Survey

The following papers were an inspiration to this project, and have therefore been listed here:

1. EAST: An Efficient and Accurate Scene Text Detector

Authors: Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, Jiajun Liang ,2017

Work Done :In this project deep learning methods were used to only detect the text in an image, the image can be natural or complex. In this it creates horizontal bounding boxes which may potentially contains the text.It can be used as text detection for text recognition.

1. DIGIT RECOGNITION USING DEEP LEARNING

Authors : Shashank Mishra,D. Malathi,K. Senthilkumar

Work Done :In this project handwritten text digit recognition on a well-known image database is done using Convolution neural network .Deep learning methods were used as it increases accuracy and reduces computation time .High accuracy is achieved using various filters which in turn gave better results.

# Problem Statement

The aim of this Project is to make an algorithm in Python that can recognize English Language text from a natural image with the help of Convolution Neural Network and Display the recognized text on the Screen.

# Objectives

We perform text extraction on an image in three independent steps:

1. Image preprocessing
2. Detection
3. Recognition.

First we process the image to fed to the network then we detect regions that potentially contain text and in the third step we perform text recognition, where, for each of the detected regions, we use a convolutional neural network

(CNN) to recognize and transcribe the word in the region.

# Methodology

1.Block Diagram



a)Workflow :

* Removal of background noises for clear and accurate reading of image and conversion of image into grayscale.
* Making of bounding boxes of text in image

i.e the area that may potentially contains the text.

* Feeding of bounding boxes from image to CNN.
* Displaying of text

a).Algorithm/Pseudo Code

#Text detection and recognition

a) Prediction of bounding box

predbound(pro\_score,geo)

(row,col)=pro\_score.shape[2:4]

box=[]

confd\_val=[]

#loop over rows for initial score data

while i :0 to

score=pro\_score[0,0,i]

x0=geo[0,0,i]

x1=geo[0,1,i]

x2=geo[0,2,i]

x3=geo[0,3,i]

angle=geo[0,4,i]

#loop for column score data

while j:0 to R

if score[j]<args["min\_confd"]:

continue

(offX,offY)=(i\*4,y\*4)

#computing of rotation angle for prediction

ang=angle[j]

#computing of dimension of the bounding box

h=x0[j]+x2[j]

w=x1[j]+x3[j]

#computing of start and end values using offX and rotating angles

#then append box start and end with confd\_val

box.append((startX,startY,endX,endY))

confd\_val.append(score[i])

return (box,confd\_val)

b)Recognition of text

# initialisation of the list of results

results = []

#result list will be filled by the region of text i.e region of interest

#loop for whole image and trying to find the coordinates of bounding box

while (startX1, startY1, endX2, endY2) in boxes:

# scaling the coordinates based on the respective

ratios in order to reflect bounding box on

the original image

startX1 = int(startX1 \* rW)

startY1 = int(startY1 \* rH)

endX2 = int(endX2 \* rW)

endY2 = int(endY2 \* rH)

#extracting the region of text or interest

r = orig[startY1:endY2, startX1:endX2]

#configuring setting for conversion of

image data to string.

configr = ("-l eng --oem 1 --psm 8")

#Recognize the text from the image of bounding box

text = pytesseract.image\_to\_string(r,config=configr)

# adding coordinates with the

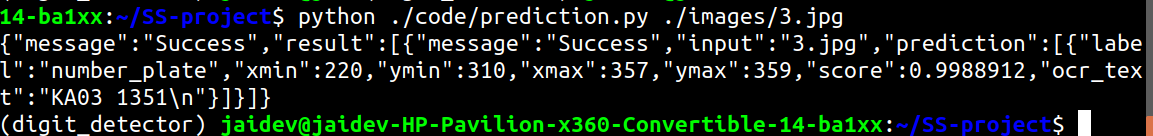
associated text to result list

results.append(((startX1, startY1, endX2, endY2),text))

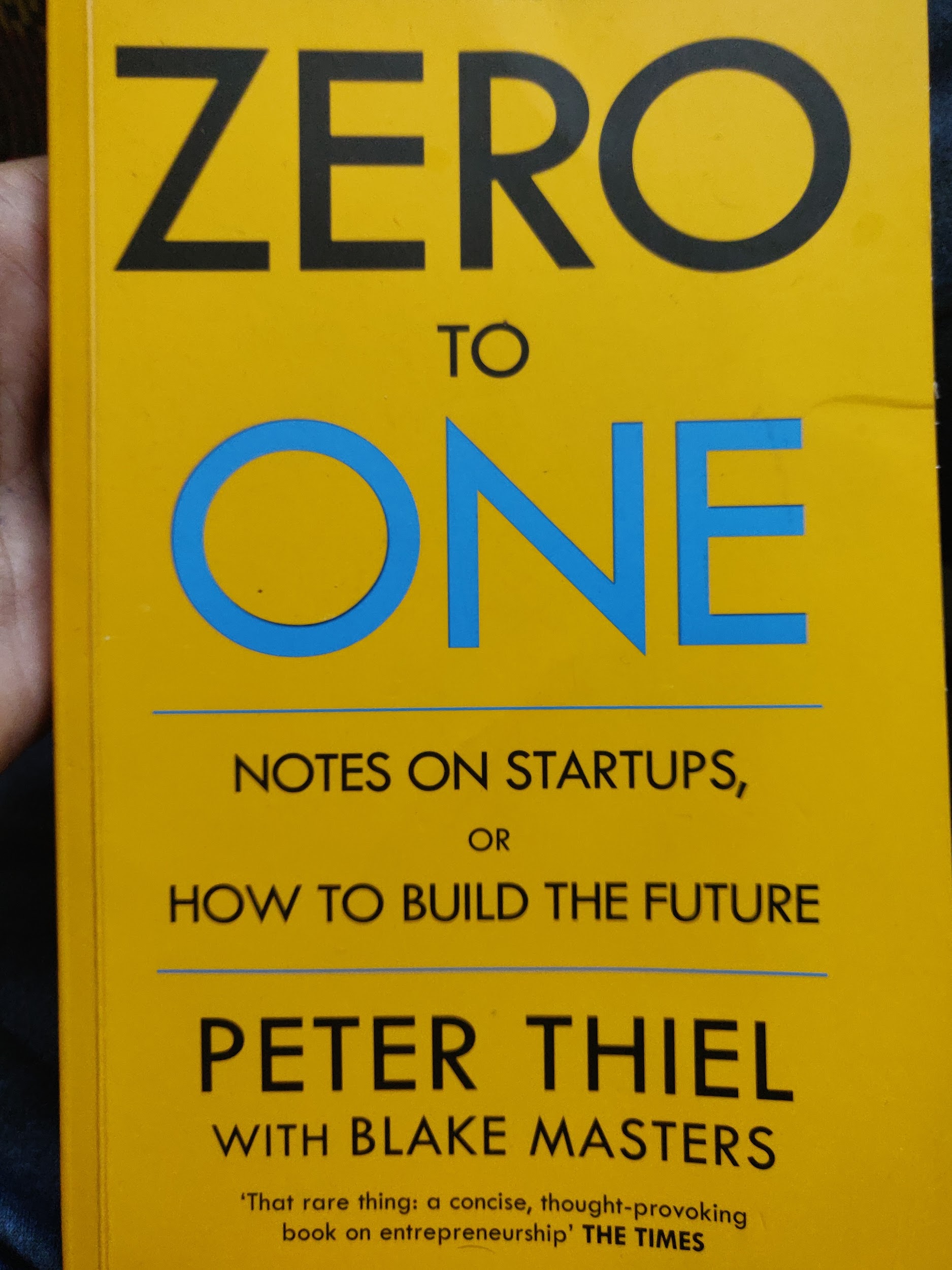
# Result and analysis

## Input image 1:

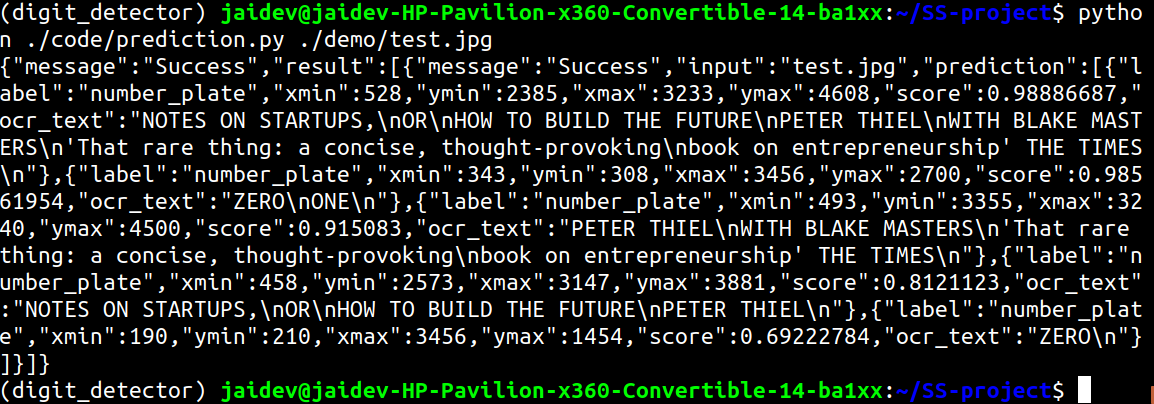
* Result :



* The output of the program gives the message that the text in the image is found i.e message = “success”.
* It shows the label of the image that where exactly is the bounding box or it can be said that the potential area where the text is found (their coordinates) : xmin,ymin,xmax,ymax.
* It gives the score for the area in the given image i.e whether it has text or not.
* Finally, it prints the text recognized in the bounding box in the image.In this case : “KA03 1351”
* Input Image 2:



* Result :



# The output of the program gives the message that the text in the image is found i.e message = “success”.

* It shows the label of the image that where exactly is the bounding box or it can be said that the potential area where the text is found (their coordinates) : xmin,ymin,xmax,ymax.
* It gives the score for the area in the given image i.e whether it has text or not.
* Finally, it prints the text recognized in the bounding box in the image.

# Conclusion

## This model is not 100 % accurate if the image is not clear and blurry the model will find it difficult to recognize the text. Still, good results have been achieved with the EAST model and CNN. Adding more filters for processing the image would help in improving the performance of the model.

##### Acknowledgment

With immense please we are presenting our report as a part of the curriculum of I.T department we wish to thank all the people who gave us unending support.

## Individual Contribution

Jaidev Chittoria (18IT119): Preprocessing and

recognition of text

Ayush Bhandari (18IT209): Detection of text and training of

data

Royce (18IT239): a Literature survey and base papers

search

##### Implemented/Base Papers

1.East:An Efficient and Accurate Scene Text Detector

Authors: Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, Jiajun Liang

Year: 2017

2.Digit detection using deep learning

Authors : Shashank Mishra,D. Malathi,K. Senthil kumar

Year:2018

X. REFERENCES

* <https://www.pyimagesearch.com/2018/08/20/opencv-text-detection-east-text-detector/>
* <https://arxiv.org/abs/1507.05717>
* <https://arxiv.org/abs/1704.03155v2>

<https://in.mathworks.com/solutions/deep-learning/convolutional-neural-network.html>