

Automate Identification and Recognition of Handwritten Text from an Image

Siddharth Salar^a, Siddhest Roy^b, Saurabh Verma^c

^{a, b, c} School Of Computing Science & Engineering, Galgotias University, Greater Noida

Article History: Received: 10 November 2020; Revised 12 January 2021 Accepted: 27 January 2021; Published online: 5 April 2021

Abstract: Handwritten text acknowledgment is yet an open examination issue in the area of Optical Character Recognition (OCR). This paper proposes a productive methodology towards the advancement of handwritten text acknowledgment frameworks. The primary goal of this task is to create AI calculation to empower element and information extraction from records with manually written explanations, with an, expect to distinguish transcribed words on a picture.

The main aim of this project is to extract text, this text can be handwritten text or it can machine printed text and convert it into computer understandable or wNe can say computer editable format. To implement this project we have used PyTesseract which is an open-sourcOCR engine used to recognize handwritten text and OpenCV a library in python used to solve computer vision problems. So the input image is executed in various steps, first there is pre-processing of an image then there is text localization after that there is character segmentation and character recognition and finally we have post-processing of image. Further image processing algorithms can also be used to deal with the multiple characters input in a single image, tilt image, or rotated image. The prepared framework gives a normal precision of more than 95 % with the concealed test picture.

Keywords: OCR, Image Processing, Handwritten Text Recognition, Artificial Neural Network, English Alphabet Recognition, Supervised Learning

1. Introduction

The project is about extraction of transcribed content from a picture, which is an optical acknowledgment of characters is the electronic or mechanical transformation of pictures of composed, manually written, or printed text into machine- encoded text, regardless of whether from a checked archive, a photograph of a record, a scene- photograph (for instance the content on signs and announcements in a scene photograph) or from caption-text superimposed on a picture.

Broadly utilized as a type of information section from printed paper information records – regardless of whether identification reports, solicitations, bank explanations, automated receipts, business cards, mail, printouts of static-information, or any appropriate documentation – it is a typical technique for digitizing printed messages with the goal that they can be electronically altered, looked, put away more minimalistically, showed on-line, and utilized in machine cycles, for example, intellectual processing, machine interpretation, (removed) text- to-discourse, key information, and text mining. OCR is a field of exploration for example acknowledgment, computerized reasoning, and PC vision.

The primary goal of this task is to create AI calculation to empower element and information extraction from records with manually written explanations, with an expectation to distinguish transcribed words on a picture.

2. Solution Approach

We're using Pytesseract using machine learning approach although there are various approaches to implement this project, at last, the main aim is to fulfill project aim and objectives[3].

To meet venture targets we have learned about Convolutional Neural Networks. The plan for picking this field since it empowers machines to see the world as people do, see it likewise, and even utilize the information for a large number of undertakings, for example, Image and Video acknowledgment, Image Analysis and Classification, Media Recreation, Recommendation Systems, Natural Language Processing, and so on the headways in Computer Vision with Deep Learning has been built and culminated with time more than one specific calculation is Convolutional Neural Network[1].

To implement this project we've used tesseract python, also we've used python tools such as OpenCV, Numpy, Python imaging library, and Pytesseract[3].

The undertaking configuration is partitioned into parts.

The initial segment intended to take the contribution of the picture whose text is to be separated.

The second piece of the undertaking is the primary piece of the venture which is intended to diminish the clamor and actualizes the Pytesseract with convolutional neural organization calculation bit by bit to distinguish the content present in the picture.

3. Assumptions

The assumptions considered are as follows:

- 1 The handwritten text must be in English.
- 2 The text across the input image must be handwritten to achieve good results.
- 3 All machine dependencies must be installed properly.

4. Project Diagrams

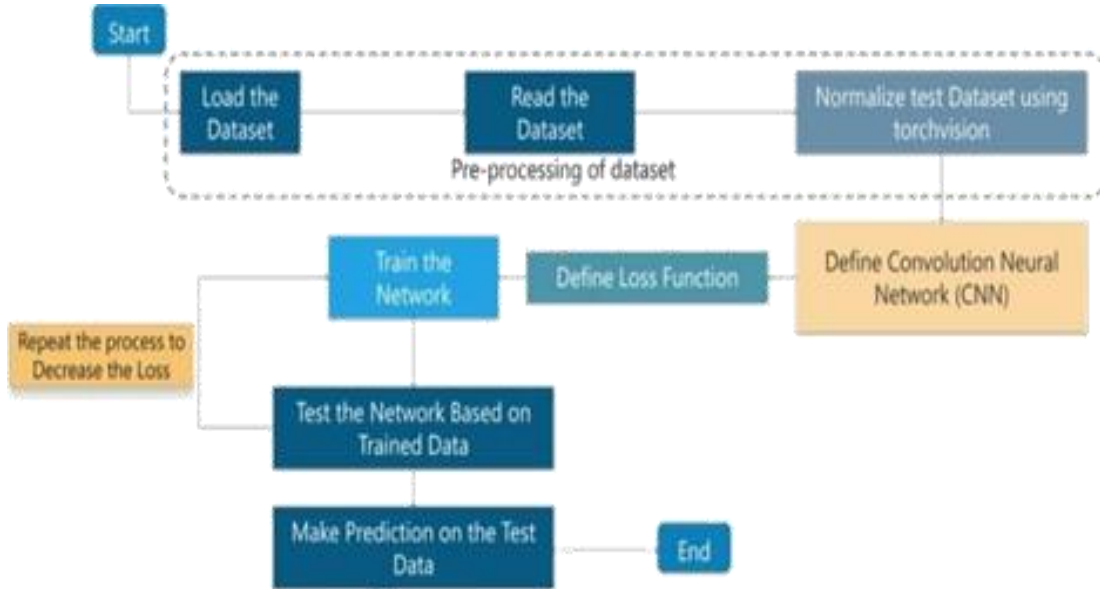


Figure 1: Logic Flow of Convolution Neural Networks, Sayantinim Deb, medium.com, Nov 27, 2018. <https://medium.com/edureka/convolutional-neural-network-3f2c5b9c4778>

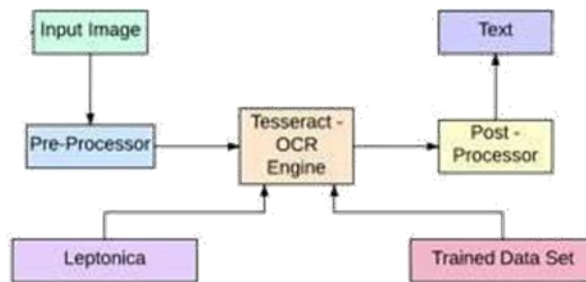


Figure 2: Logic Flow of Pytesseract OCR Engine, Google.com

5. Algorithms

The algorithm used to implement the project is the Pytesseract OCR engine which uses a convolutional neural network algorithm that is used by the tesseract optical character recognition engine in python[3]. There are four layered ideas we ought to comprehend in convolutional neural networks:

- 1 Convolution
- 2 Rectified Linear Unit
- 3 Pooling Layers
- 4 Fully Connected Layer
- 5 Convolution of an Image

Convolution has the pleasant feature of being translational invariant. Naturally, this implicit that every convolution channel speaks to a component of the premium (e.g pixels in letters) and the Convolutional Neural Network calculation realizes which highlights contain the subsequent reference (for example letter set)[2].

There are 4 stages for convolution:

- Line up the component and the picture
- Multiply each picture pixel by comparing highlight pixel
- Add the qualities and discover the total
- Divide the total by the absolute number of pixels.

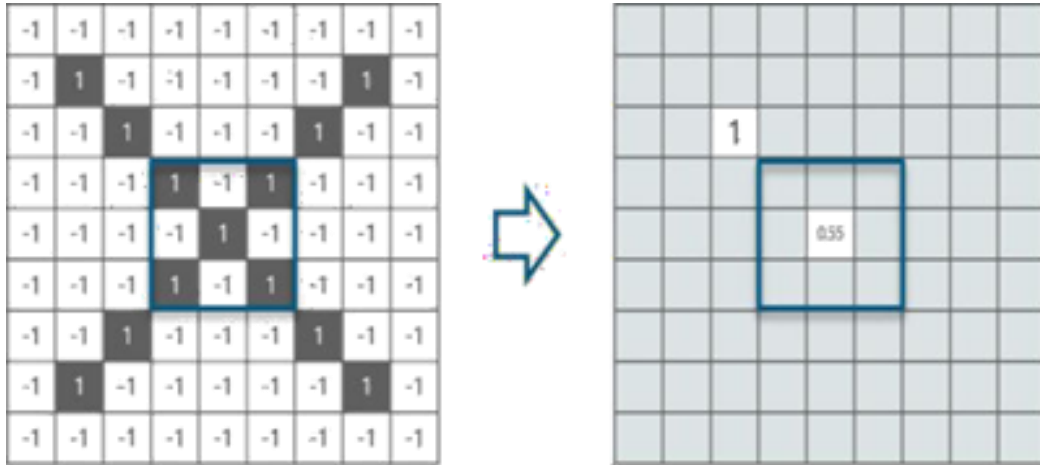


Figure 3: Convolution of an Image, Sayantinim Deb, medium.com, Nov 27, 2018. <https://medium.com/edureka/convolutional-neural-network-3f2c5b9c4778>

The yield signal strength isn't reliant on where the highlights are found, yet essentially whether the highlights are available. Thus, a letter in order could be present in various locations and the convolutional neural network calculation, at present have the option to remember it.

Rectified Linear Unit

Change work possibly initiates a hub if the info is over a specific amount, while the information is under zero, the yield is zero, however, when the info transcends a specific limit, it has a straight association with the needy variable[1].

The principle point is to eliminate whole negative qualities from the convolution.

Positive qualities continue as before however the negative qualities are converted to zero as demonstrated as follows:

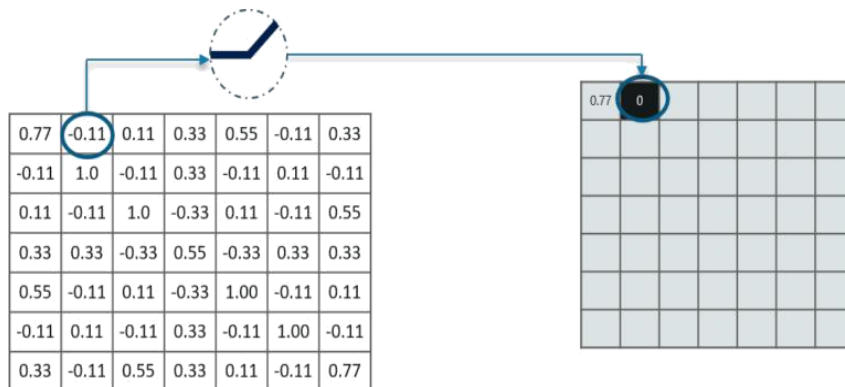


Figure 4: Rectified Linear Unit, Sayantini Deb, medium.com, Nov 27, 2018. <https://medium.com/edureka/convolutional-neural-network-3f2c5b9c4778>

Contributions from the convolution layer must be leveled to decrease the affectability of the channels to commotion and varieties. This leveling cycle is known as sub testing and must be accomplished by identifying midpoints or by finding the most extreme over an example of the sign.

Pooling Layer

In Pooling layer the therapist the picture stack into a more modest size. Pooling is done in the wake of going through the enactment layer[1]. This can be done by executing these 4 steps:

- Select a window size (generally 2 or 3)
- Select a step (generally 2)
- Move the window across your sifted pictures
- Select maximum value from each window.

We select the size of the window as 2 and we got 4 qualities to browse. From these 4 qualities, the greatest incentive present is 1 so we pick 1. Likewise, we began with a 7x7 lattice yet now a similar framework in the wake of pooling boiled down to 4x4.

Yet, we need to get the window across the whole picture. The strategy is as same as above and we need to rehash that for the whole picture. Do take note that this is for one channel. We need to do it for 2 different channels too.

This is done and we show up at the accompanying outcome:

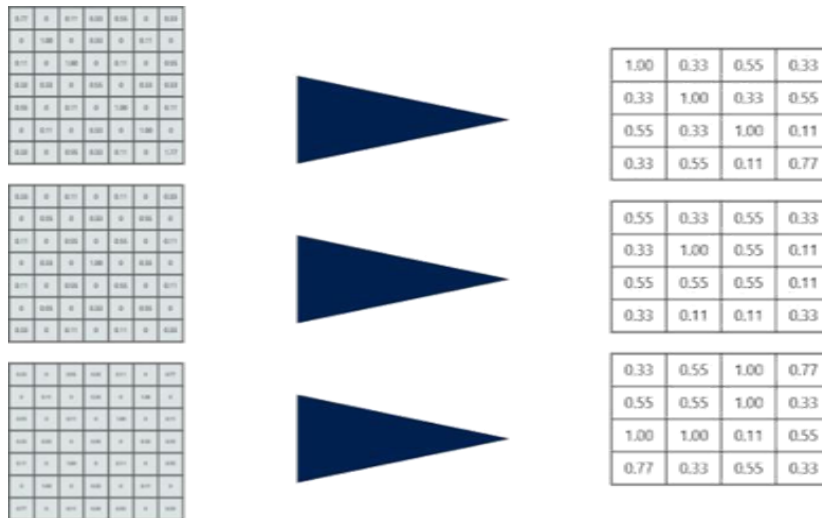


Figure 5: Pooling Layer, Sayantini Deb, medium.com, Nov 27, 2018. <https://medium.com/edureka/convolutional-neural-network-3f2c5b9c4778>

This process is over, in the next step we will do the stacking of the layers!

Stacking Up the Layers

In order to bind the time period in a single picture here we've a 4x4 framework from a 7x7 framework after going the contribution across 3 layers – Convolution, Rectified Linear Unit, and Pooling[2].

We further diminish the picture from 4x4 to 2x2 to accomplish this we have to play out all 3 tasks in emphasis after main pass. Therefore, subsequent to the second pass, we show up at a 2x2 framework as demonstrated as follows:

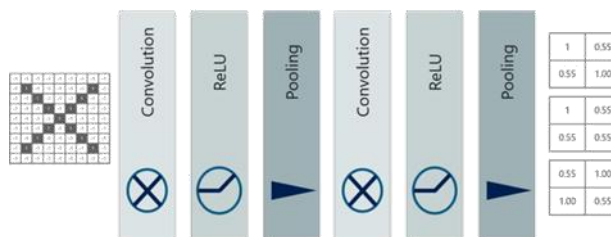
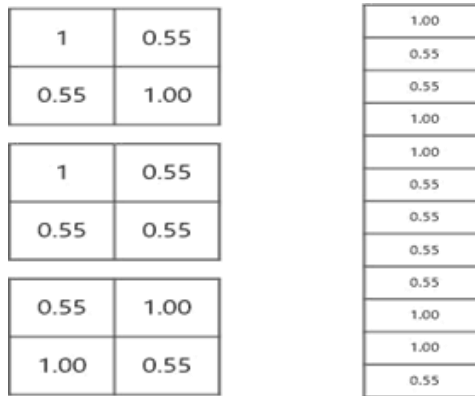


Figure 6: Stacking Up the Layers, Sayantini Deb, medium.com, Nov 27, 2018.
<https://medium.com/edureka/convolutional-neural-network-3f2c5b9c4778>

In the network rearmost layers are completely associated, implying that neurons of going before layers are associated with each neuron in resulting layers.

This impersonates significant amount of thinking where all potential pathways from the contribution to yield are thought of. Likewise, a completely associated layer is the last layer where the order occurs[2]. Now we take our separated as well as yelled pictures and dispose them into a single rundown as demonstrated as follows:



Now, when we provide in, 'X' and 'O', some components in the vector will become high.

Observe the picture beneath, as should be obvious for the values of 'X' there are various components that are large, and correspondingly, for the values of 'O' there're various components that are large:



All things considered, what did we comprehend from the given picture is the point at which the first, fourth, fifth, tenth, and eleventh qualities are high; we can arrange the picture as 'x'. The idea is comparative for different letter sets too – when certain qualities are orchestrated in the manner in which they will be planned to a genuine letter or a number which is required[2].

Prediction of Image Using Convolutional Neural Networks –Fully Connected Layer

Now, we're finished preparing the network and now, we can start to foresee and look over the functioning of the classifier[4]. We should look at a basic model:

We got a 12 component vector got after the course of the contribution of an arbitrary letter into and out of all the layers of our organization.

We make forecasts dependent on the yield information by contrasting the got qualities and rundown of 'x'and 'o'.

0.9
0.65
0.45
0.87
0.96
0.73
0.23
0.63
0.44
0.89
0.94
0.53

Figure 7: Prediction of Image Using Convolutional Neural Networks – Fully Connected Layer, Sayantini Deb, medium.com, Nov 27, 2018.

We just added the qualities we discovered as high (first, fourth, fifth, tenth, and eleventh) from the table (vector) of X and persuaded the aggregate to be 5[2]. We did precisely the same thing with the info picture and got an estimation of 4.560.

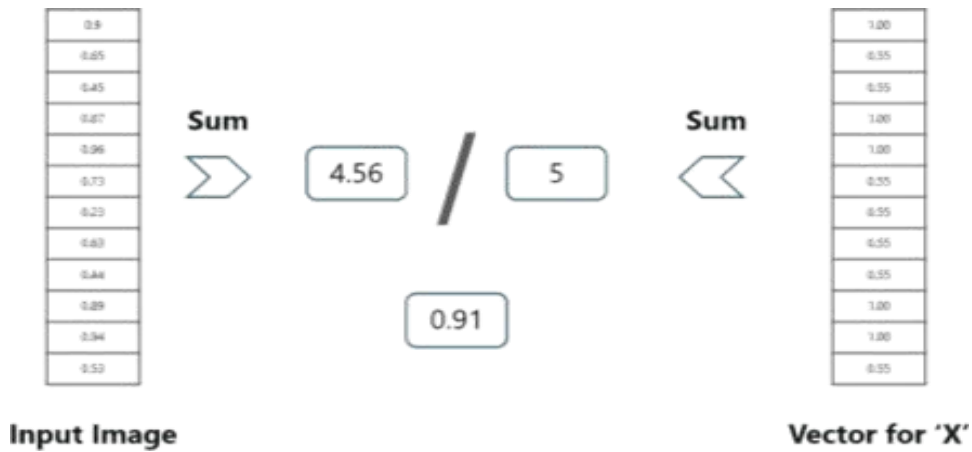


Figure 8: Vector table, Sayantini Deb, medium.com, Nov 27, 2018.

<https://medium.com/edureka/convolutional-neural-network-3f2c5b9c4778>

At the point when we partition the worth, we've a likelihood match, i.e. 0.91! How about we do likewise with the table (vector) of 'o' presently:

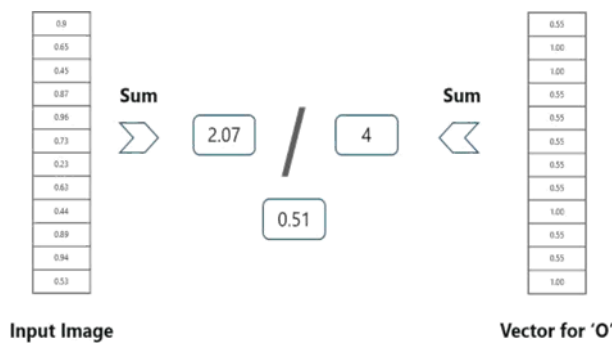


Figure 9: Vector table, Sayantini Deb, medium.com, Nov 27, 2018.

<https://medium.com/edureka/convolutional-neural-network-3f2c5b9c4778>

We have the yield as 0.51 from this table. Indeed, the likelihood being 0.51 is under 0.91, right?

So we can infer that the subsequent info picture is an 'x'. Furthermore, this is the manner by which forecast work is finished.

6. Outcome

The calculation can identify and section manually written content from a picture. The model is effectively ready to identify the greatest words in a given line of sentence or words, which makes it about 90% precise while execution and testing.

For instance, the info picture having the manually written content is given as follows:



Figure 10 : Input Image to the OCR, Google.com

The model processes the image removes the noise from the image and the Pytesseract performs the convolution neural networks and predicts the text[4].

Extracted Text: Jim Morrison

As we can see the model is quite accurate and successfully able to extract the handwritten text.

The model predicts and extracts the text from the image as follows:

```
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
# Recognize text with tesseract for python
result = pytesseract.image_to_string(Image.open(img_path))

# Remove template file

return result

print ("____The Recognition Process Of Text From The Input Image Started Successfully____")
print ("The Recognized Text From The Input Image IS:")
print (get_string(filename))

print ("____Process Finished____")
____The Recognition Process Of Text From The Input Image Started Successfully____
The Recognized Text From The Input Image IS:
Jim Morrison
____Process Finished____
```

Whereas another image having the handwritten text is given as follows:

The model processes the image removes the noise from the image and the Pytesseract performs the convolution neural networks and predicts the text[4].

Extracted Text: This is the handwritten

Example

Write as qoool as you can

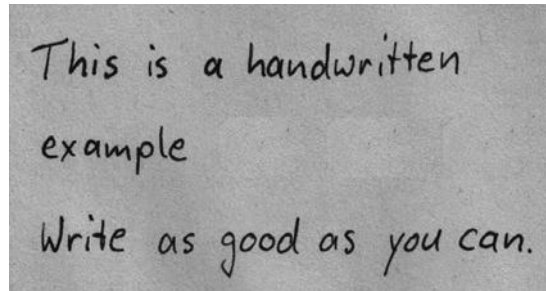


Figure 11 : Input Image to the OCR, Martin Guggisberg, Jan 2006, Research Gate, https://www.researchgate.net/publication/224759117_Multimedia_Information_and_Mobile-Learning/figures?lo=1

As we can see the model is quite accurate and successfully able to extract the handwritten text.

The model predicts and extracts the text from the image as follows:

```
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
cv2.imwrite(img_path, img)

# Recognize text with tesseract for python
result = pytesseract.image_to_string(Image.open(img_path))

# Remove template file

return result

print ("____The Recognition Process Of Text From The Input Image Started Successfully____")
print ("The Recognized Text From The Input Image IS:")
print (get_string(filename))

print ("____Process Finished____")

____The Recognition Process Of Text From The Input Image Started Successfully____
The Recognized Text From The Input Image IS:
This is a handwritten
example

Write as good as you can.
____Process Finished____
```

7. Exceptions Considered

The exceptions considered are as follows:

- 1 The text across the input image must be of the same color, not multicolor handwritten text.
- 2 The image doesn't have too aggressive multicolor backgrounds across the text of the image.
- 3 The image doesn't have any kind's objects in the background across the text of the image.

8. Enhancement Scope

The enhancement scope of this project is following:

- 1 The accuracy of the model can be increased with predefined models and powerful machine learning GPU processors can be used to attain a good percentage of accuracy.
- 2 In the future, we can use this algorithm with more than one particular language.

References

Rokas Balsys, "Introduction to Convolutional Neural Networks", May 8, 2019. <https://medium.com/analytics-vidhya/convolutional-neural-networks-cnn-explained-step-by-step-69137a54e5e7>
Sayantini Deb, "Developing a Image Classifier in Python", Nov 27, 2018. <https://medium.com/edureka/convolutional-neural-network-3f2c5b9c4778>

Filip Zelic & Anuj Sable, “A Comprehensive guide to OCR using Tesseract”, Jan 2019.<https://nanonets.com/blog/ocr-with-tesseract/>
Adrian Rosebrock, “Tesseract OCR: Text localization and detection”, May, 2020.
<https://www.pyimagesearch.com/2020/05/25/tesseract-ocr-text-localization-and-detection/>
Karez Abdulwahhab Hamad, Mehmet Kaya, “A Detailed Analysis of Optical Character Recognition Technology”, International Journal of Applied Mathematics, Electronics and Computers, September 2016.