International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 08 | Aug 2020 www.irjet.net p-ISSN: 2395-0072

OPTICAL CHARACTER AND DIGIT RECOGNITION SYSTEM

Aditya Panchwagh[1], Mayank Modi[2]

[1-2]*Student, Dept. of Computer Science and Engineering, MIT School of Engineering, MIT ADT University, Pune, Maharashtra, India*

Abstract - *Since olden times, the need for storing information in various ways has always been there. This was very useful until we felt the need to reuse this information again and again. In request to reuse these snippets of data, we had to read and search individual contents from different documents and then rewrite it again. Thus, there is an explicit need for automated softwares or programs in order to provide fast and accurate methods to revive the text from the long-lasting images and documents.*

Also, as we are advancing in this world full of technology, where new softwares are emerging, there is an increasing demand for softwares that recognize characters or words which can be converted into editable documents used wherever and whenever we want. As we surf the internet, we come across various facts, figures or data that we need, but cannot be replicated or edited since these documents are collected in the form of images. In our busy lives, we need everything instantly. Hence, the data being small or large, typing it can be time consuming. In such scenarios, the software like the Optical Character Recognition comes into play.

In this research paper, we have proposed the system which detects text from images or documents and converts them into an editable format which is more efficient and less time consuming. We also have a developed a system that detects numbers which are obscure and displays them in an understandable format. We used OCR algorithms, python libraries and ML pre-processing concepts and we used it on datasets that give us significantly good outputs with an efficient and simple manner.

***Key Words*: OCR, Machine Learning, Python, CNN, Pre-Processing.**

1. INTRODUCTION

OCR is like a combination of eye and mind of human body. Every time a human being reads, the eyes and brain implement optical character recognition without even realizing it. Human brains have the ability to recognise and understand the patterns or words that they read. Thus, the brain perceives the data and this is the way we apply OCR. However, a computer being a machine, does not have any eyes or brain, so how can OCR be performed? In this scenario, we need to feed the computer with pictures of the data using a camera, rather than the text itself.

Thus, OCR is a character or pattern recognition technique which uses algorithms to recognize and extract characters of a word and display them in an editable format making it more efficient for the user. OCR can be achieved by online as well as offline character recognition. Now in both the ways the non-editable text is converted into editable text, however in online format it is done simultaneously as the letters are drawn.

In our project, the aim was to create a system that could detect and extract characters from an image or a document which were not clear or which were not editable. For that, we used the OCR algorithm with python libraries like OpenCV and NumPy to scan the input image, clear it and then print it so it can be used easily. Along with that, we created a system that could recognize the digits given to it as inputs and correctly classify it as output. We used the machine learning algorithms to train the MNIST dataset and perform classification operations on it.

2. RELATED WORK

Text detection and recognition is not a new problem at all. We are trying to solve this problem for decades now. We are still trying to find a solution using different methods which are more efficient and has less computational cost. To solve this problem, many researchers have tried and given lots of different approaches in the past. Here we present a detailed study of some interesting approaches proposed by researchers to solve the problem of recognizing the text from images.

Manwatkar in his research discussed about text recognition from images. In that, he proposed to use Kohonen neural network, which just contains input and output layer, means no hidden layers. Error rate in this proposal is very high, hence, it doesn't give accurate results.[1]

Gur proposed a text recognition algorithm, which works with use of fuzzy logic rules that further uses the statistical data of different texts. They focused on just one type of font and were unable to recognize the text which was written using different fonts in image.[2]

Rahiman discussed about using OCR method for recognition of the old south Indian scripts from the images. But, using his OCR system one can recognize the text which is just in form of South-Indian languages.[3]

Karthick in his project used third generation OCR systems. But these systems don't perform well when image is of poor quality and not clear.[4]

Hamad in his research work discussed about, how we can easily recognize the text from the provided image, but his system suffers from multiple drawbacks, such as, when image is blur or distorted, or when there is little skewness present in image or text present in image. Also, it does not perform well when different fonts were given as an input.[5]

Marne in his research paper suggested use of Tesseract, which is basically an OCR engine, to extract the text from the input image. But Tesseract suffers from a drawback that, the input provided image should be of high resolution and clear, so that it can extract very accurately.[6]

3. METHODOLOGY

3.1 Optical Character Recognition System Pipeline

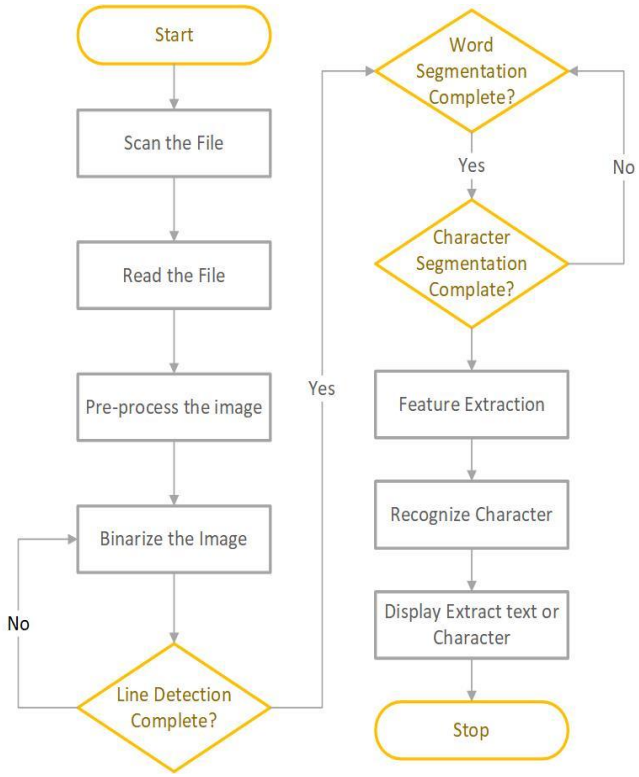

Fig -1: OCR work-flow

[1] Image Pre-processing

The most important problem in OCR lies in image pre-processing. In this part, the image undergoes a lot of changes so that the quality of the given image can be improved. It consists of processes like Binarization, where the coloured image is converted into black and white image, then aligning the image so as to get only the necessary part and removing all the unnecessary part, then removing noise to make the image smother to work on. These are the steps for image pre- processing.

[2] Segmentation and Feature Extraction

In the segmentation phase, the input text is broken into smaller parts i.e. the image is first segmented into lines, then each line is divided into words and finally each word is again divided into characters.

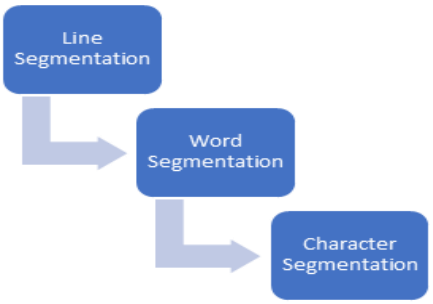

Fig -2: Segmentation Process

Then the feature extraction process is carried out which then matches the segmented characters with the characters in the database and classifies them correctly. Generally, the problem arise in this phase since this process is entirely dependent on how the classification is done for the characters. However, training the dataset using different machine learning algorithms is the key to get a good accuracy.

[3] Recognition

In this phase, the database for characters is trained using machine learning algorithms so that the maximum accuracy can be achieved. The actual decisions for the classification of a character are made here. After the character is classified successfully, it is then displayed, which is in an editable format.

[4] Dataset

A dataset consists of collection of information of similar items. The dataset we used for our ocr algorithm is MSRA Text Detection. It consists of 500 natural images which were captured by normal camera.

3.2 Digit Recognition System Pipeline

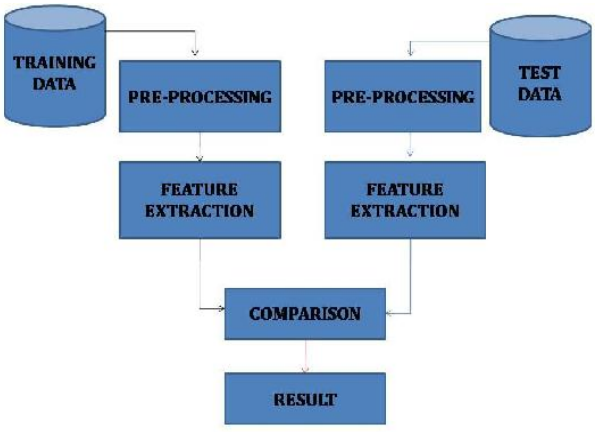

Fig -3: Digit Recognition work-flow

[1] Dataset

We here used, the famous MNIST dataset to train our Convolutional Neural Network (CNN). MNIST dataset

consists of sixty-thousand small square of 28X28 pixel grayscale images of handwritten digits from 0 to 9. We've trained our CNN on this MNIST dataset and then we used our model for digit recognition.

[2] Pre-Processing

The image with digit written, is given as input to the CNN. But first we need to pre-process the image, by first converting it to grey scale image, as dataset we've used to train our model contains only grayscale images. After converting it to grayscale, we'll scale our image so that it can be easily matched with trained data and can be accurately recognized by our model, that too using less memory and less computation time.

[3] CNN Model

Our CNN model consists of ten layers. As, we have already pre-processed our image, we directly have grayscale image as input to our model. Hence when our model will create matrix of pixels of the image, it will be in form of 0X3X3, because we have grayscale image and another pixel will form matrix of 3X3.

We have used the activation function as relu during pooling phase and we have used pooling of size 2X2, means we have converted our 3X3 matrix into 2X2 using pooling, that too without losing any data of image. Pooling is just like compressing a matrix. As we know that main role of CNN is to reduce the image in a certain form so that, it is easier to process without losing features which are important for getting the prediction accurate.

At, output phase we have used sigmoid activation function. Also, we have used optimizer as 'adam' and testing for 10 epochs. If we increase the epochs above 10 then it will be of no use as it doesn't improve the as much.

4. APPLICATION

[1] Captcha

Captcha is a renowned system used in websites to distinguish humans from machines. During captcha there is a verification process to recognize and replicate the distorted letters in the blank box. However sometimes it happens that the captcha given to us is very vague or obscure and we cannot identify it, resulting in a loss of time as well as energy. Here our OCR software comes into play.

However, when the image (fig -4) was passed through our OCR algorithm, it was detected perfectly and we could edit and paste it in the output box. Also, if the captcha contains digits with characters, it can be deciphered accurately. This is the advantage of using OCR softwares, so that we do not have to spend our time and energy decoding images that can be done in fraction of seconds by these softwares.

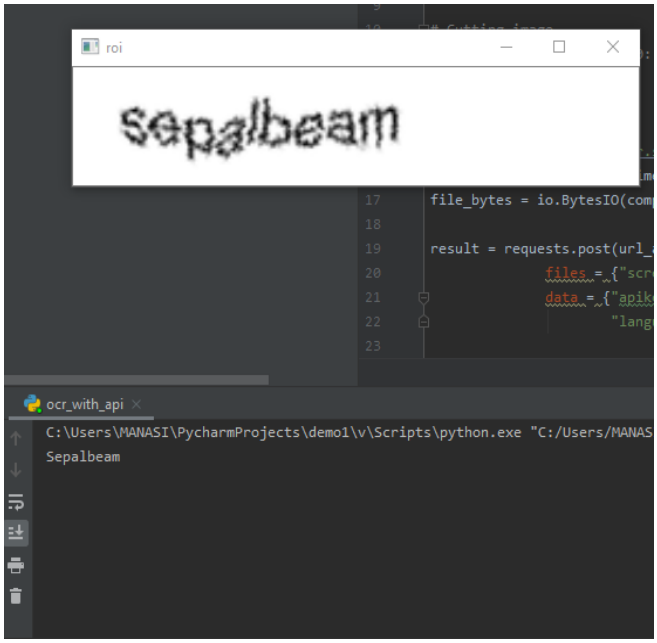

Fig -4: Captcha Input-Output

[2] Editability

It is one of our system's another application where the text from a given input image can be extracted and edited. Since many times we find the suitable information within scanned documents like images, we cannot replicate them, thus OCR softwares help up in editing text from such documents.

[3] Reduced Search Time

Once we extract the huge text from the given source image, the next important aspect is searching the required information from that decoded document. So by saving those decoded files as .docx or .txt format, we get an inbuilt search option through which we can search and edit only necessary information rather than writing it from an non-editable image document.

[4] Converting written Script into Digital Format

With the help of OCR, we can simply scan the page of a book, an article from the newspaper and record it digitally for further use. This scanned text can be edited and copied using the OCR technology.

For example: While making presentations, we gather various articles related to our subject from different books. The question here is how do we use them in our presentation work? The answer is simple- using Optical Character Recognition. With OCR, we can simply scan those article pages from the books or we can also simply click pictures of them and convert those images in an editable format so that there is no need to store and maintain these books when we can digitalize them and use them easily.

5. RESULTS

5.1 Output of Character Recognition System

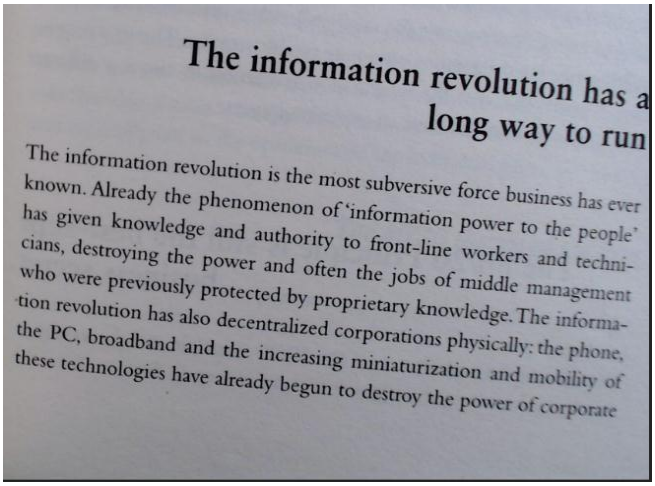

Fig -5: OCR Input

On giving input image of a book as shown in fig.5 to our system, we got the output as shown in fig.6 by using OpenCV and NumPy modules. Thus, our system is able to recognize and extract the vague or blurred part of image even when it is slightly rotated at certain angle, and that too with good accuracy.

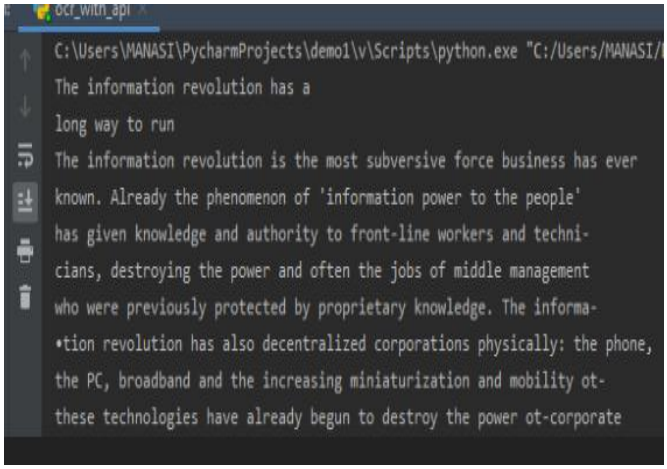

Fig -6: OCR Output

In this way OCR is very useful to edit such information that we get from books and store them digitally so we can use them efficiently.

5.2 Output of Digit Recognition System

After training our model, when we give input as image number 99, which is labelled as 1 in MNIST dataset, Our model will match it's pixel matrix with other matrices present in dataset and predict the number which is written in image and provide output as 1, which is shown in output below. After prediction we got our model's accuracy as 99.02%.

Fig -7: Digit Recognition Input-Output

5.3 Accuracy graph

This accuracy graph is for our digit recognition system which shows that, during training and testing our model, if we increase the epochs, then accuracy of training and testing of model increases up to a certain extent.

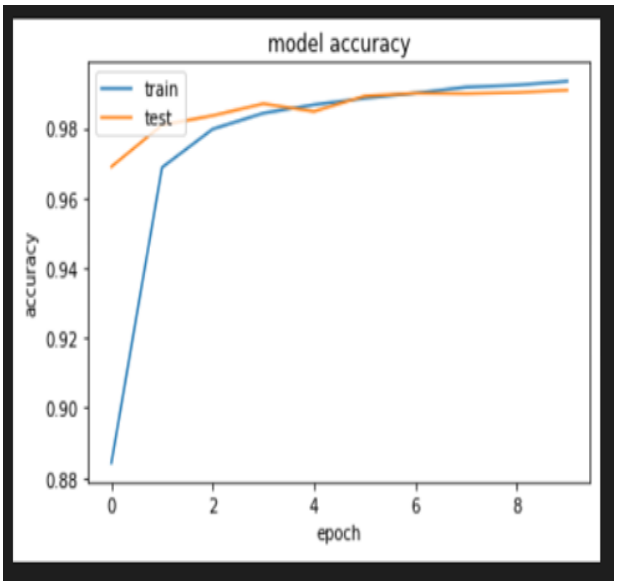

Fig -8: Accuracy Graph

5.4 Loss graph:

The loss graph in fig.9 is for our digit recognition system which shows that, when we train or test our model and increase epochs, loss of data decreases, but after certain limit we can't reduce loss.

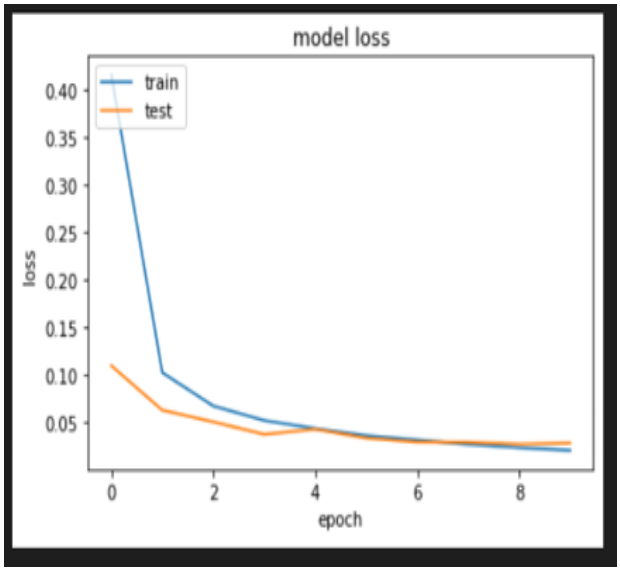

Fig -9: Loss Graph

6. FUTURE WORK

Currently our system for OCR and Digit Recognition is API based, so in future we tend to make it an application which when published on different social media platforms can be useful to many users to digitalize their data.

Further we intend to upgrade our system by adding a handwriting recognition feature in it so that users can not only scan printed text but also can use our system to scan handwritten text and store that information with them.

We can combine our system with Brain Computing Interface (BCI). When we think, our brain produces certain types of waves, and using BCI we can catch those waves and can see about those on computer machine. We can then apply our system to that to convert those into natural language, which we humans can read and interpret easily.

7. CONCLUSION

The objective for this research paper was to create a system that could detect and extract text from images or scanned documents which can be converted into an editable format with maximum accuracy. OCR is a technology that eases the use for humans in terms of searching and editing documented information and helps save a lot of time as well. Throughout this system, we could successfully use a variety of python libraries lie OpenCV, NumPy for scanning the images and processing them, then segmenting them and matching it with our database to classify them correctly and using Machine Learning algorithms like CNN for digit recognition by cross referencing it with the widely used MNIST dataset so as to get the maximum accuracy for obtaining better classification results.

8. REFERENCES

[1] Gur and ZeevZelavsky, "Retrieval of Rashi Semi-Cursive Handwriting via Fuzzy Logic", IEEE International Conference on Frontiers in Handwriting Recognition (ICFHR), 2012.

[2] A. Hyvarinen and E. Oja, "Independent component analysis: algorithms and applications", Neural networks, vol. 13, no. 4-5, pp.411-430,2000.

[3] Phillips D 2000 "Image Processing in C" R & D Publications Lawrence, Kansas, USA.

[4] Mori, M 2010 "Character recognition" Sciyo Publisher Croatia.

[5] M. A Hasnat, M. R Chowdhury, M. Khan, "Integrating Bengali script recognition support in Tesseract OCR".

[6] Michael Hadourin "A Neural Network Implementation of Optical Character Recognition" Technical Report Number CSSE10-05 COMP 6600 – Artificial Intelligence Spring 2009.

[7] Mahmoud, S.A., & Al-Badr, B., 1995, Survey and bibliography of Arabic optical text recognition. Signal processing, 41(1), 49-77.

[8] Teofilo E. de Campos, Bodla Rakesh Babu, Manik Varma, "Character Recognition in Natural Images", International conf. on Intelligence Science and Big data Engg., pp. 193-200, 2011.

[9] Cherneta, DS, Druki, AA &Spitsyn, VG 2016 "Development of multistage algorithm for text objects recognition in images" International Siberian Conference on Control and Communications (SIBCON), Moscow, pp. 1-5.

[10] R. Bradford & T. Nartker. Error Correlation in Contemporary OCR Systems. Proceedings ICDAR-91, Vol. 2, p. 516-524, 1991.

[11] Shreya Dutta, Naveen Sankaran, Pramod Sankar K, C.V. Jawahar, "Robust Recognition of Degraded Documents Using Character N-Grams", IEEE,2012

[12] Dr. Neetu Bhatia, Optical Character Recognition Techniques: A Review, International Journal of Advanced Research in Computer Science and Software Engineering.

[13] Matteo Brisinello, Matija Pul, Tihomir Andelic, Ratko Grbic, Improving Optical Character Recognition performance for low quality images, 59th International Symposium ELMAR-2017,18-20 September 2017.

[14] Chaudhari A., Mandviya K., Badelia P., K Ghosh S., Chapter 2, Optical Character Recognition Systems, Optical Character Recognition for different Languages with soft computing, 2017, XIX 248 p, 95 illius, Hardcover.

[15] Amarjot Singh, Ketan Bacchuwar, and Akshay Bhasin, A Survey of OCR Applications, International Journal of Machine Learning and Computing, Vol. 2, No. 3, June 2012.

[16] Zongyi Liu, Ray Smith, A Simple Equation Region Detector for Printed Document Images in Tesseract, 2013

12th International Conference on Document Analysis and Recognition.

[17] Huimin Lu, Baofeng Guo, Juntao Liu, Xijun Yan, A Shadow Removal Method for Tessearct Text Recognition, 2017 10th International Conference on Image and Signal Processing, Biomedical engineering and Informatics.

[18] Optical Character Recognition using Neural Networks Deepayan Sarkar University of Wisconsin Madison ECE 539 Project, Fall 2003.

[19] "A Neural Network Implementation of Optical Character Recognition" Technical Report Number CSSE10-05 COMP 6600 – Artificial Intelligence Spring 2009.

[20] Ye Q, Doermann D. Text detection and recognition in imagery: A survey. IEEE transactions on pattern analysis and machine intelligence. 2015 Jul 1;37(7):1480-500.

[21] Verma R, Ali DJ. A-Survey of Feature Extraction and Classification Techniques in OCR Systems. International Journal of Computer Applications & Information Technology. 2012 Nov;1(3).

[22] Suen CY. Character recognition by computer and applications. Handbook of pattern recognition and image processing. 1986:569-86.

[23] Lund, W.B., Kennard, D.J., & Ringger, E.K. (2013). Combining Multiple Thresholding Binarization Values to Improve OCR Output presented in Document Recognition and Retrieval XX Conference 2013, California, USA, 2013. USA: SPIE.

[24] Satti, D.A., 2013, Offline Urdu Nastaliq OCR for Printed Text using Analytical Approach. MS thesis report Quaid-i-Azam University: Islamabad, Pakistan. p. 141.

[25] Arica, N & Vural, FTY 2001, "An overview of character recognition focused on off-line handwriting" IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 31, no. 2, pp. 216-233.

[26] R. Gossweiler, M. Kamvar, S. Baluja, "What's Up CAPTCHA? A CAPTCHA Based on Image Orientation", in WWW, 2009.