# Character recognition using tesseract enabling multilingualism.

NIHARIKA, G.L, BANO, S., KUMAR, P.S., DEEPIKA, T. and THUMATI, H.

2020

# *Character Recognition using Tesseract enabling Multilingualism*

Gorsa Lakshmi Niharika
Department Of CSE
Koneru Lakshmaiah Education Foundation
Vaddeswaram,India
niharikagorsa2000@gmail.com

Shahana Bano
Department Of CSE
Koneru Lakshmaiah Education Foundation
Vaddeswaram,India
shahanabano@icloud.com

Pavuluri Shyam Kumar
Department Of ECE
Vellore Institute Of Technology
Chennai,India
pavulurishyam55@gmail.com

Tinnavalli Deepika
Department Of CSE
Koneru Lakshmaiah Education Foundation
Vaddeswaram,India
deeputinnavalli1999@gmail.com

Hampi Thumati
Department Of CSE
Koneru Lakshmaiah Education Foundation
Vaddeswaram,India
hampi.sps@gmail.com

*Abstract*— **Character recognition builds a recognizing factor for identifying the accuracy in characters. The accuracy of classifying the recognizing characters in an image is applied through deep learning methods. The character recognition is mainly focusing on the layers of text recognition through deep learning techniques. Well cleared python code assists to furnish all the levels of image by following deep learning that algorithmically analyse and recognize text from the given input image. This research work has been proposed for recognizing characters using deep learning techniques and recognize the input image with well-furnished and most efficient output. It provides a high level of accuracy-built output after the recognition of characters in the high-resolution image. This recognized character can be converted into user desired languages where the proposed model is trained to recognize some particular languages.**

***Keywords— Character recognition, Tesseract, deep learning, OpenCV, Tkinter, images, multi lingual translation.***

## I. INTRODUCTION

At present, the research on deep learning techniques for processing and recognition of characters can be implemented with some inbuilt opensource packages.

Characters play a key role in research literature. These characters give life to the feeling of words. For recognition of these characters in an image is a task for the machine. By using deep learning methods, we categorize the text in an image.

To go through in research methodology, the entire research is implemented in the python language. The python code makes ease for conversion or recognition of text elements with an accuracy of prediction.

The characters are classified with the inbuilt opensource package called Tesseract, which classifies the text elements. Here, in this model that takes input from the user for classifying and recognizing the text or characters present in the image. Optical character recognition (OCR) [8] is also a part in the character recognition that makes the difference, where OCR deals with the electronic recognition of characters in the handwritten materials [7] [13] [14] [15], pdf files, word files, and many text containing documents. These hidden layers of characters classification are built with deep learning algorithms. While the system can't learn to detect the text with ease of running the code. It needs to train and test all the characters present in the English alphabets. Whereas, the recognition of text in an image is a task where the machine needs to analyze the characters by training through each layer of deep neural networks. Here, in the recognition of characters in an image, it has some noises within the image. These are removed in multiple levels or layers of deep learning methods.

The image is scaled and it is exposed to remove some extra noises involved in the image. In this model, these are classified by applying dilation and erosion to remove some extra noises and holes in the images. These make recognition of text with ease of output. When grey scaling is performed to the images [5] it can remove the false classifications and complexities with an accuracy of prediction [3].

So, when we give input to the machine it takes an image and processes the data for recognition of text in the inputted image. Here in this recognition using Tesseract, Tkinter, and Open CV, deep learning algorithms. To recognize the characters, an open-source package that has inbuilt libraries and it is developed by Google. The noise removal in an image makes ease to calculate and classify the characters in an image. The deep learning algorithm helps to iteratively learn the model and recognize the characters with the accuracy in prediction. Some algorithms fail in recognition of elements in an image that gives an output of desired and accuracy as given in an image.

The deep learning algorithm trains the image to do scaling and classify accordingly for the recognition using Tesseract OCR [13]. This tesseract works as a character recognition package, when one tries to install it. Deep neural networks [2] makes character classification with a high resolute approach in output.

The saturated output is stored in a temporary variable, function calls for converting the data into desired language. Here in our model we trained our system with 4 languages i.e., German, Dutch, English, Telugu. So, when the user

gives the path of the image, the model gives us a language selection pop up window.

Although there are many existing models which recognizes the text apparently. Here in this research, deep learning algorithms were processed in providing with more efficient accuracy.

## II. Related Work

For the ultimate point of our research study, we had gone through all the existing Research models [4] and practical applications [6]. The unique implementation of Tesseract OCR cannot generate accurate results. Through the numerous iterations of deep neural networks and Open CV, algorithms [9][10][11][12] produce accurate results when compared with the previous one. From the Wikipedia of Optical character recognition, we carefully analyzed the consistent data into two effective ways of the proactive approach of recognition of characters. One is a matrix matching that accepts images and checks from a pixel by pixel. This distinct type of approach instantly makes to recognize well-written documents with known fonts. Another type of approach recognizes the preprocessed and recognized characters in every iteration of evaluating characters.

There have been many methods to classify the text or characters from an image but here in our research methodology we thought our research work should give a classification and accuracy in the output to be obtained from the image. When one uses deep learning techniques in the findings of text it gives ground truth value after recognition of text in an image.

The analysis to recognize the handwritten character recognition [1] using the Modern National Institute of Standards and Technology (MNIST) database [16]. These characters written through hands are recognized in multiple layers of deep neural network algorithm. The handwritten images in common were grey scaled to instantly recognize without any distinct noise and false classifications. Conversely, in our extensive research, we typically took some quoted images as input it carefully removes the false classifications and noises in the image. When the raw data is given as input the code cannot generate accurate text present in the image. So to classify the right text in the image, we used deep learning methods.

Furthermore, in detail of the existing approach of taking Tesseract as only one to recognize the characters are difficult. The main con approach in recognition methods is due to the dull background, misrecognition of fonts. The classification function of recognition is failed in this case of approaches.

A more effective way from our findings can help out, by considering the deep neural networks architecture. The deep neural networks can be augmented through removing invariance achieved in the incorrect output. It is a coherent approach not to accept a raw input directly for the recognition of characters. Therefore, it has to be greyscale, apply dilation and erosion to eliminate noises. This unique way of effective implementation is discussed briefly in the proper procedure.

The main of this work is to identify the way to recognize the characters in an image and in later stage it will convert into different languages desired by the user.

| Method | Approach |
|---|---|
| Tool for recognition | Tesseract OCR |
| Algorithm included | Deep Learning |
| Dilation and erosion | Removed noises |
| Accuracy | 98% |
| Integration | Conversion recognized characters into desired language. |

Table.1: Overview of model

The novelty of proposed research work is integrating the Tesseract OCR and enabling multilingualism of recognized characters.

## III. Proposed Methodology

### *A. Importing all the packages :*

To run our model, we need to import some required packages which are openly available to get install for the recognition of characters in an image. Import the path for tesseract in order to call the recognition of characters function in later stage. In the same way of importing required packages and libraries, import image path for providing the machine to recognize characters in an image. This character recognition needs Tesseract package which was developed earlier by Google. Through this approach of recognition of characters are preprocessed with all the existing characters. When there is a noise and dull picture for identification of the characters in an image that can be solved by deep neural networks. Deep neural networks algorithm makes efficient in eliminating the noises and the image is gray scaled accordingly we can detect the text in the input image.

For the classification and pattern analysis of the characters are calculated by NumPy package. NumPy solve the mathematical approach in the code. Here for our understanding the output is gathered and shown in a message box. The message box is obtained through the implementation and import of Tkinter. This gives an effective way of output in a pop-up window.

Open CV is a deep learning technique where we mainly used to execute the data with calculating after many eliminations of noises. Open CV is the most popularly used algorithm in the branch of deep learning techniques.

### *B. Deep Learning Algorithm:*

Deep learning algorithm works in the mainly hidden layers of finding the result. The networking's of taking input from the user and iterates in the loop of recognition. Deep learning algorithms makes the more of structured and unstructured data into the correct, accurate result of the output. The least recognition of text can lead to incorrect identification of characters. Of course, this better validation of text and making to get the output is highly appreciable algorithm. With the help of Tesseract engine, the recognition of text in an image makes the identification accurate.

Here deep learning techniques makes the colored image into greyscale picture as it gives identification for engine to an extent level. The application of deep learning algorithm helps to eliminate denoise in the images up to greater extent.

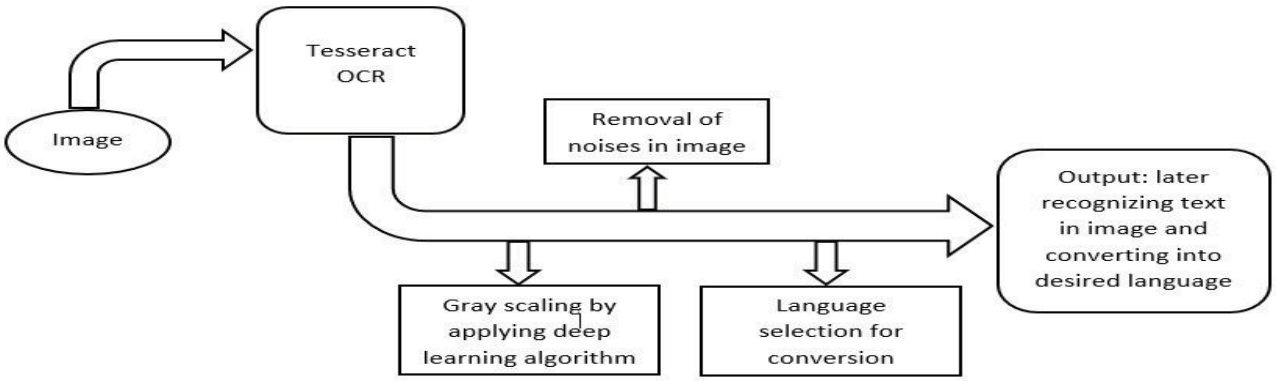


Fig.1: Block diagram of proposed method

By using deep learning algorithms, the discussed changes obtained for a high-resolution image. In this method the feature extraction is obtained for the input image to understand the text written in image. This algorithm makes use of features by predicting labels recurrently in the image.

*C. Working Approach:*

If one wants to, do the analysis with the text present in the image can't be retrieved in the swift of seeing image. There are many online sources where we drop an image gives in the output of text present in image. But it cannot give us accurate result of output through the image. Virtual characters present in the image can't be identified clearly by the Tesseract engine. So, hereby we are using the data to be preprocessed for the recognition of text present in the image. The preprocessing algorithm do analysis with the characters existing in it.

The problem factors for affecting the recognition of characters in an image are given in detail below:

- The foremost drawback to recognize the text in an image is noise. The noised image can give false assumptions for identification of characters. If these are more in an image then training of characters is much needed through every iteration of loop. So, to ensure this draw back in the image we used deep learning algorithms to eliminate those noises.
- If the taken image is not in correct position or not in correct angle to identify the text in image.
- Reflection of the characters gives incorrect results in identification.
- Not all fonts make correct assumptions for recognition of text. This also makes false beliefs of characters.

Considering these all problems occurred in recognition of text, we are proposing in our research work to take deep learning algorithm. This algorithm can help to remove the extra unwanted noises in the image and it makes to scale the picture. So that no unwanted or problem making things won't occur for identification or recognition of text. Gray scaling of the picture or an image makes to exit the unnecessary belief for color complexion. Light mismatches at some areas can also make the false belief, so through grey scaling it makes correct and accurate predictions.

Image is cleaned in high resolution state for further recognition process of tesseract engine. Dilation and erosion are occurred to remove unwanted noises for better resolution. Whereas dilation adds pixels for boundaries to process the characters in an image, erosion removes the pixels on object boundaries in processing an image.

When it makes all the perfect and clear blue water recognition or prediction of exact words, we take that as our output. Else it goes on like iterative process until it accords accurate value. These outputs were displayed using tkinter package for better understanding and visualization result or recognition of text. So, this deep learning algorithm produces recognized text for a gray scaled, noise removed image.

**Step 1:** Start.
**Step 2:** The packages and all the required engines to be import. Here for our code of execution we imported NumPy, Tesseract, Tkinter packages.
**Step 3:** When we imported these all packages and engine, import an image which needed recognition of text in image.
**Step 4:** If the image has more colour complexion or any light mismatches the deep learning algorithm Open CV eliminates those by converting the image into greyscale image.
**Step 5:** The algorithm of deep learning checks for any unwanted noises present in the image. If it is more noised image then it removes all those noises and present the high-resolution image after adapting threshold.
**Step 6:** Input image is calculated to brief view of text identification and the pre-processing of the characters is done through Tesseract engine.
**Step 7:** If the text is not identified or pre-processed correctly iterate the steps recurrently from step-1.
Else continue.
**Step 8:** Select the language to convert the text.
**Step 9:** Displays the text.

**Step 10:** Click Ok to finish with the recognition.
**Step 11:** Repeat the steps 1,2,3,4,5,6,7,8.
**Step 12:** End.

*D. Input Analysis:*

For the better understanding of this application of deep learning one has to get a clear note of packages importing these were required packages and engines where one have to install in the system and import them accordingly.

The tesseract is an online engine which was developed by Google needed to be installed in our system and that path should be given correctly. The main aim of this code is to give input an image. So, for that set out the path of an image into your local system.

After providing the image and engine path we have to allow the OpenCV algorithm to read the image and it should perform the conversion normal colored picture to grey-scaled one.

Apply the dilation and erosion to eliminate the unwanted noises which are present in the image. Within a glimpse of iteration, a system can't identify and predict the character. So, to recognize and classify the characters present in the text were identified accordingly, immediately the removed noise picture was saved locally in our system.

Thresholding of the image is a type segmentation done for any image. This makes the converted grey-scale image and noise removed image to form a binary image. It accords a very high resolute image for allowing to next step of recognition of characters using Tesseract engine.

The recognized characters will be the input of conversion of text into different forms. Here in our model google translator which makes to change the data into desired language. Languages that are trained in our model are German, Dutch, Telugu, and English.

Although the image characters were recognized by using with help of tesseract it has some challenges in identifying the correct characters. It by alone can't give accurate results in character recognition. In recognition process if the image consist of more noises gives not adequate value of output. If tesseract engine built with Machine learning algorithms like LSTM then output efficient is more in analyzing the characters. LSTM algorithm works with learning past values it occurs recurrently for learning new characters. It will be really helpful to learn new style of characters by using Tesseract including neural network system based on LSTM.

*E. Flow chart:*

The flow we represented here in flowchart is started with an algorithm of importing all the packages and required google developed engine for character recognition. Tesseract makes an ease for this process of execution. We give input to the code as image with any extensions it gives a recognized output through many iterations. Output is prompted like a pop-up box where it displays the recognized characters. First the input image was inserted into the code, that is converted to the greyscale image. The image contains many noises which includes false assumptions of the characters. So, it is necessary to convert the image into high resolute image for future more classification. Estimate the lines obtained in the image for recognition of text. Through this technique it draws assumptions for the outline of characters. Classifies the characters based on the outlines and takes the gaps into consideration of assumption. Matches the output with the desired one else it will repeat the loop from starting to get accuracy in the result. If it matches then it continues with the recognised assumptions and displays it in the pop-up box. The display panel appeared after running the code is language selection, where one has to opt his desired text translation. The overview of our model is shown completely in Fig 2.

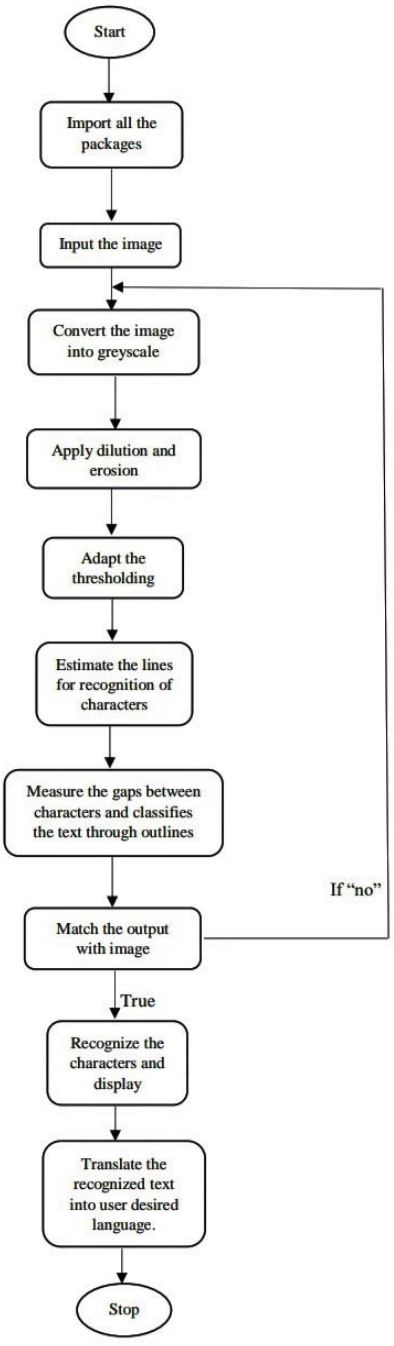


Fig 2: Overview of the process

## IV. RESULTS

After many iterative assumptions here we get the desired outputs for our built model. The input was taken from the user and do analysis with the image we provided.

Fig 3 is image which we given for the input analysis of text.

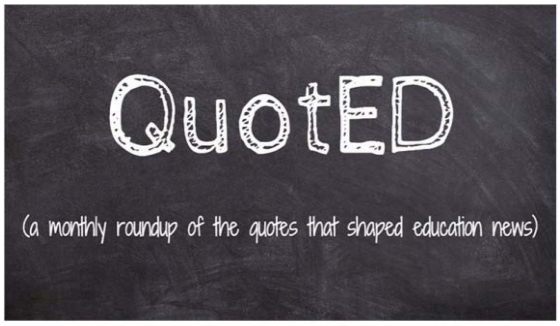


Fig 3: Input image given in code for assumption of text.

Fig 4 after running our model we get the language selection panel where we selected in each iteration of image character recognition. Calling the Tkinter function formulates the GUI poping up, to select the desired languages.

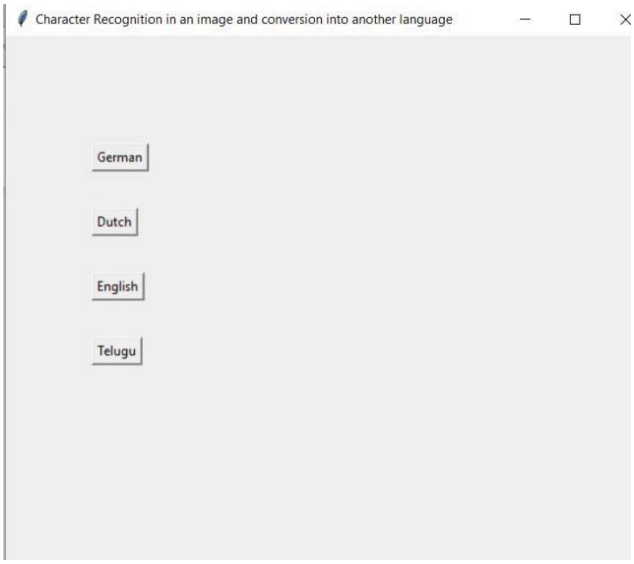


Fig 4: Language Selection Panel after every iterative step of code.

After selection of Language, characters recognized will translate accordingly into selected language. Here Fig 5 resembles the same after selecting Telugu as the translating language for Fig 3 image.

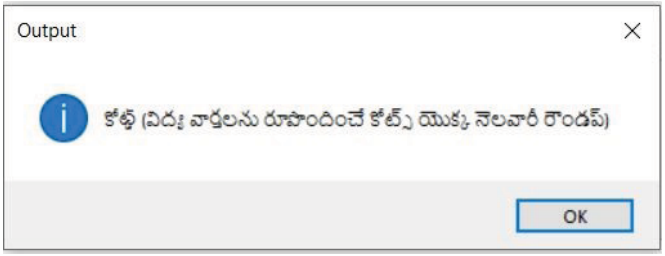


Fig 5: Output of the input image taken from user after selecting Telugu.

These output images obtained through the multiple iterations of the code we gave. Here with respect of the font the image or picture, we can recognize if it is clear and high resolution for recognition of text or characters in it. But through when we make the image to greyscale image then it gives a correct recognition capability to identify characters.

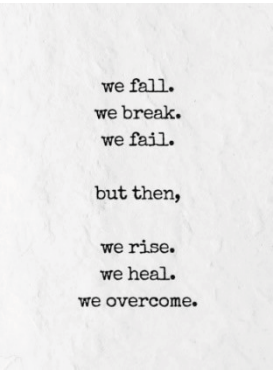


Fig 6: Input of the image given by user


Fig 7: Output of the image taken from user after selecting Dutch.

The same way of approach processing was observed with different set of images. So, Fig 6 and Fig 8 are the inputed images for recognizing the characters and translating into desired language. Fig 7,9 are the images after translation of characters. User have selected to translate Fig 6 into Dutch language. So the output resembled in dutch language. Same for the Fig 8 is selected to translate into german after recognizing the characters in an image.


Fig 8: Input of the image given by user

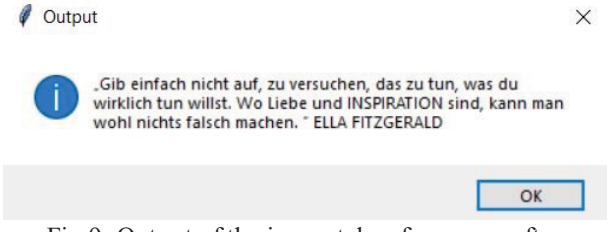


Fig 9: Output of the image taken from user after selecting German.

The image literally includes all the unwanted noises, blurriness, lack of resolution and light complexions. These

were removed with an iterative approach of deep learning techniques. Fig 10, shows if an image is given by user it is converted into grey scale formatted image. Apply dilation and erosion technique where it resolves all the unwanted sounds in the grey scale image.


Fig 10: Input of grey-scale image.

After grayscaling the image apply deep learning algorithm to remove unwanted noises (Fig 11). After remvoing these noises in an image it will be easy for the Tesseract engine to deal with characters and their differentions.


Fig 11: After removing unwanted noises in the image.

Through all these approaches of iterations the final output is displayed in the Fig 12. Quality of these images become hints for the recognition of text or characters in the image. Another thing to note is our results shows the iterative analysis of recognition through classification and calculation the outlines of characters.


Fig 12: Output of the input image taken from user after selecting English.

| Metrics | Accuracy |
|---|---|
| Designed images | 98% |
| Plain texted images | 99% |
| Style fonted text images | 95% |

Table 2: Validations with different images

Differentiating with various types of images and comparing its accuracy (Table 2). When the color variations are more in an image then it is difficult for machine to eliminate all the noises and other disturbances in identifying the text. Our algorithmic approach given accuracy mentioned in Table 2.

Here is a limitation which we done thoroughly, through online tools. This tool takes input image from the user and it got converted into text format. By which the tool identifies the text in an image and displays it as output. Similarly, we took an image to recognize the text available in it. But it fails in some cases for recognizing the text due to the problem factors that effecting in recognition. So, soon the output displayed is in error format that is discussed in below image Fig 13. The tool fails to identify the text and displayed as "No recognized text!".

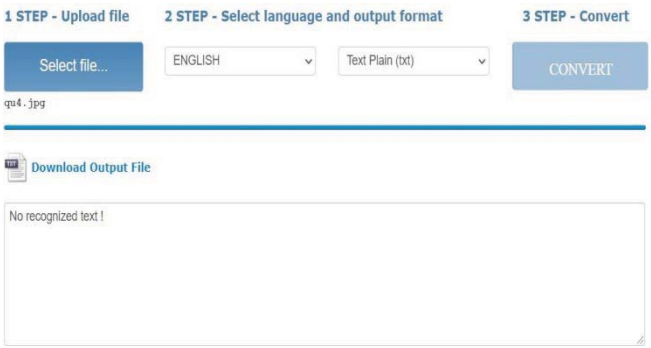


Fig 13: Error source from the online report analysis.

The received output will be given as input for language translation when the characters are recognized in first stage of process.

## V. CONCLUSION

This analysis model can make text recognition in all type of formats when the input image is provided by the user for recognition. Moreover, the packages and engines used for recognition makes an ease to deploy the character recognition. Tesseract and deep learning combination will make the model a highly effective approach for translation or conversion of identified text to display as output. This technique is used widely for the extraction of data present in an image. This model also consumes less time for character identification. Open CV and Tesseract are used widely in this research for character recognition.

## VI. FUTURE SCOPE

For future enhancement of our research this can be used at the educational purpose which keeps on the records of data of students. These taken inputs such as certificates and mark sheets [9] of students can be stored into database.

As deep learning algorithm makes an ease for recognition of characters present in the images of datasheets of students. One can develop this technique at many places in recognition of characters for example in shops, bank accounts etc.,

## REFERENCES


[1] S. Acharya, A. K. Pant and P. K. Gyawali, "Deep learning based large scale handwritten Devanagari character recognition," 2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), Kathmandu, 2015, pp. 1-6, doi: 10.1109/SKIMA.2015.7400041.

[2] Oyebade K. Oyedotun, Ebenezer O. Olaniyi, Adnan Khashman, "Deep Learning in Character Recognition Considering Pattern Invariance Constraints," 2015 I.J. Intelligent Systems and Applications, 07, pp. 1-10.

[3] Morgan McGurie, "An image registration technique for recovering rotation, scale and translation parameters", Massachusetts Institute of Technology, Cambridge MA, 1998, pp.3.

[4] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In ACM Transactions on Graphics (TOG), volume 23, pages 689–694. ACM, 2004.

[5] Tongbo Chen, Yan Wang, Volker Schillings, and Christoph Meinel. Grayscale image matting and colorization. In Proceedings of Asian Conference on Computer Vision, pages 1164–1169. Citeseer, 2004.

[6] Thiyagarajan, S. & Kumar, G. & e, Praveen & Sakana, G.. (2018). Implementation of Optical Character Recognition Using Raspberry Pi for Visually Challenged Person. International Journal of Engineering and Technology (UAE). 7. 65-67. 10.14419/ijet.v7i3.34.18718.

[7] Rushikesh Laxmikant Kulkarni, "Handwritten Character Recognition Using HOG, COM by OpenCV & Python", International Journal of Advance Research in Computer Science and Management Studies, Volume 5, Issues 4, April 2017.

[8] Optical Character Recognition Technique Algorithms, N. Venkata Rao, Dr. A.S.C.S. Sastry, S.N. Chakravarthy, Kalyanchakravarthi P, JATIT paper, 15 vol 83 NO 2.

[9] Velasco J.S. et al. (2020) Alphanumeric Test Paper Checker Through Intelligent Character Recognition Using OpenCV and Support Vector Machine. In: Beltran Jr. A., Lontoc Z., Conde B., Serfa Juan R., Dizon J. (eds) World Congress on Engineering and Technology; Innovation and its Sustainability 2018. WCETIS 2018. EAI/Springer Innovations in Communication and Computing. Springer, Cham. https://doi.org/10.1007/978-3-030-20904-9_9.

[10] Sweta Kumari, Leeza Gupta, Prena Gupta, "Automatic License Plate Recognition Using OpenCV and Neural Network", International Journal of computer science trends and technology, Vol 5, Issue 3, 2017, pp. 114-118.

[11] Jain, Pratiksha, Neha Chopra, and Vaishali Gupta. "Automatic License Plate Recognition using Open CV." International Journal of Computer Applications Technology and Research 3.12 (2014): 756-61.

[12] R. Vaidya, D. Trivedi, S. Satra and P. M. Pimpale, "Handwritten Character Recognition Using Deep-Learning," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, 2018, pp. 772-775, doi: 10.1109/ICICCT.2018.8473291.

[13] Andrew S. Agbemenu, Jepthah Yankey, Ernest O. Addo, "An Automatic Number Plate Recognition System using OpenCV and Tesseract OCR Engine" International Journal of Computer Applications, Volume 180 – No. 43, May 2018. pp. 1-5.

[14] M. A. Wibowo, M. Soleh, W. Pradani, A. N. Hidayanto and A. M. Arymurthy, "Handwritten javanese character recognition using descriminative deep learning technique," 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Yogyakarta, 2017, pp. 325-330, doi: 10.1109/ICITISEE.2017.8285521.

[15] A. Ashiquzzaman and A. K. Tushar, "Handwritten Arabic numeral recognition using deep learning neural networks," 2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR), Dhaka, 2017, pp. 1-4, doi: 10.1109/ICIVPR.2017.7890866.

[16] Baldominos, Alejandro & Sáez, Yago & Isasi, Pedro. (2019). A Survey of Handwritten Character Recognition with MNIST and EMNIST. Applied Sciences. 2019. 3169. 10.3390/app9153169.

[17] Sethy, A., Patra, P. K., & Nayak, S. R. (2018). Ripplet transformation based off-line handwritten character & numeral recognition. Journal of Advanced Research in Dynamical and Control Systems, 10(5), 31-39.

[18] Meghana, P., Sagar Imambi, S., Sivateja, P., & Sairam, K. (2019). Image recognition for automatic number plate surveillance. International Journal of Innovative Technology and Exploring Engineering, 8(4), 9-12.

[19] Venkata Rao, N., Sastry, A. S. C. S., Chakravarthy, A. S. N., & Kalyanchakravarthi, P. (2016). Optical character recognition technique algorithms. Journal of Theoretical and Applied Information Technology, 83(2), 275-282.

[20] Rao, G. Ananth; Kishore, P.V.V., Selfie video based continuous Indian sign language recognition system, AIN SHAMS ENGINEERING JOURNAL, December 2018, volume 9, index 4, pp.no: 1929-1939.