Document Digitization using Optical Character Recognition - A Case Study Of Mark Entry Automation in Educational Institutions.

Author 1
Institution

Author 2
Institution

Author 3
Institution

Author 4
Institution

Author 5
Institution

***Abstract*—This paper analyses and provides solutions to alleviate the challenge of manual mark entry in educational institutions by introducing a user-friendly software tool that combines Optical Character Recognition (OCR) technology and advanced artificial intelligence (AI). This tool allows educators to capture images of handwritten digits on answer scripts using a camera, converting them into a CSV file for efficient data management. Leveraging cutting-edge frameworks like TensorFlow and a custom-built Convolutional Neural Network (CNN) model, the results indicate that the project achieves precise digit recognition and outperforms existing solutions. Beyond education, this modular system holds potential for broader applications in streamlining data handling and reducing manual labor in various domains.**

***Index Terms*—Optical Character Recognition (OCR), Comma Separated Values(CSV), Artificial Intelligence (AI), Convolutional Neural Network (CNN), Artificial neural networks (ANN)**

I. Introduction

The rapid advancement of Artificial Intelligence (AI) has revolutionized various aspects of human workflow, particularly in the field of data processing. One prominent application is the automation of handwritten digit recognition [9], which holds immense potential for numerous purposes. While recognizing digits from a small set can be relatively straightforward, dealing with vast quantities of handwritten digits poses significant challenges, as it becomes a time-consuming task prone to errors.

To address this critical issue, the project aims to develop a robust and reliable system capable of instantaneously converting handwritten digits into structured Comma Separated Values(CSV) files. By doing so, it alleviates the substantial burden placed on educators during the manual data entry process. This innovative system will empower teachers and researchers to allocate their valuable time more efficiently towards other essential responsibilities, ultimately enhancing the efficiency and accuracy of digit data processing in various domains.

II. Literature Survey

The literature review has provided a comprehensive analysis of various research studies and works relevant to the proposed system. These studies have yielded valuable insights and multiple potential solutions for addressing each phase of the system's development.

Firstly, a thorough examination of OCR technology [1] has equipped us with a deep understanding of its historical context, current state, and potential future advancements. This analysis was instrumental in selecting the most appropriate OCR technology for integration into the system, considering the strengths and limitations of various OCR techniques to ensure accurate and efficient digit recognition from scanned answer script images.

Furthermore, research overviews on table extraction [10] and table detection [20] methods have significantly contributed to the system's capability to detect and extract table structures from scanned answer scripts. By exploring various approaches and algorithms, these studies have offered valuable insights into effective methods for identifying and capturing tabular data, enabling the system to accurately process and interpret tables, which are often crucial in academic and assessment-related documents.

Moreover, a comparison study conducted in [11] and [15] played a pivotal role in determining the optimal approach for implementing the system. By evaluating Long Short-Term Memory (LSTM) and CNN models, this research highlighted the unique strengths of each method and their suitability for different aspects of the system. Consequently, it provided the necessary knowledge to make informed decisions on how to leverage both LSTM and CNN effectively within the proposed system, ensuring enhanced performance and accuracy in various processing tasks.

In addition, discussions on the applications of CNN and LSTM in different contexts [4], [5], and [6] further corroborated the suitability of the chosen approaches for the system. Analyzing how these models have been successfully deployed in diverse scenarios showcased the versatility and efficacy of CNN in

handling image-related tasks and reinforced the decision to integrate it into the system.
Lastly, the focus on Python pickling and unpickling methods [14] has offered insights into an efficient means of serializing and deserializing Python objects. Drawing from this study, the suggestion to utilize Flask API for building the user interface of the proposed system emerged as a pragmatic approach. The lightweight and versatile nature of Flask API, along with its compatibility with Python's pickling methods, makes it an ideal candidate for providing a user-friendly interface to interact with the system.

III. System Description

The system is divided into 3 main modules as given in figure 1.

Fig. 1. Main components of the system

The system uses a camera to accept images as input as per the requirements. This obtained input is then processed using the img2table library. Using this library, the table structure (that is, cell coordinates) is extracted and cropped to obtain the values inside the cell. These values are then given to the TensorFlow OCR model [13] to accurately classify and predict the values obtained from the cell extraction step. The obtained result from this step is arranged specially in the format of a CSV file.
The explanation for each block is as follows:

1) Image Input: Users can edit this module to make the system work with any input type like PDF, at the end the system should work on a single image at a time.
2) OCR Model: To make the system more robust to different handwritings this module can be edited or replaced with a better model weights file.
3) Table Processing: Users can edit this module to process tables of different formats (rows and columns).

IV. Case Study - Marks2CSV

This case study highlights a transformative project aimed at alleviating the challenges faced by educators in manual data entry of student exam marks. The project leverages AI and machine learning to create an automated system that extracts data from answer scripts and converts it into standardized CSV files. The primary motivation behind this initiative is to free teachers from time-consuming data entry tasks, allowing them to focus on critical aspects of teaching, such as personalized student support and professional development. Additionally, automation enhances operational efficiency by reducing the risk of errors in student records, enabling informed decision-making and curriculum improvements within educational institutions. This case study underscores the positive impact of technology-driven solutions in the education sector, benefiting both educators and institutions alike.

A. Data Collection

Initially, 614 images of data were collected privately. Understanding the fact that this data was insufficient to train the model properly, a collection of an additional 21,600 images from a public Kaggle dataset was downloaded. As per the project requirements, the removal of image classes - numbers 8 and 9 from the public dataset reduced the public dataset image count from 21,600 images to 17,454 images. Hence the final dataset has a sum of 18,068 images.

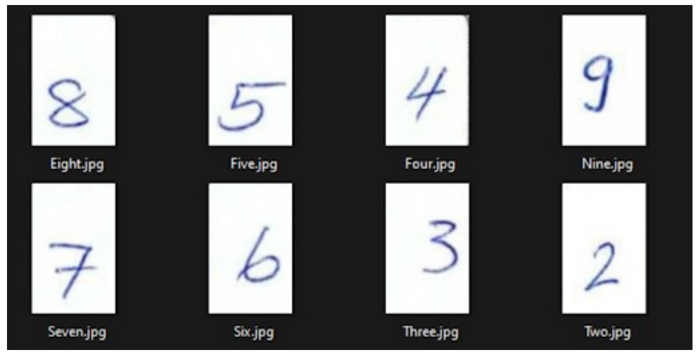


Fig. 2. Images of number in the public dataset from Kaggle

B. Methodology

The main methods in the discussion are the Convolutional Neural Network (CNN) model which is designed exclusively for mark detection purposes, and the processing of the table to the final output.
1. The CNN Model: The model architecture includes three convolutional layers with 16 filters of size 3x3 and ReLU activation, and they are coupled with max pooling layers to reduce the spatial dimension, except the last CNN layer which is a standalone CNN layer without max pooling layer. Refer Figure 3.
2. The table processing phase: The system accepts the images of answer scripts through a camera. The answer script used in the work is the answer scripts of the instituiton, the format of the answer scripts is arranged as a horizontal table that consists of question number written on the first row (13 cells), and the first column is a place-holder for headings (Qn. No. and Total), and possible divisions of questions in each question number (A, B, C). Overall, the table has 65 total cells where 36 cells are mark cells.
Method 1: img2table library
The method employs the img2table library, incorporating PaddleOCR for table detection, cell coordinate extraction, and OCR processing. These processes are automated through an input image attribute, which is converted into the library's document type. However, this approach lacks fine-grained control over individual stages for customization. Afterward,

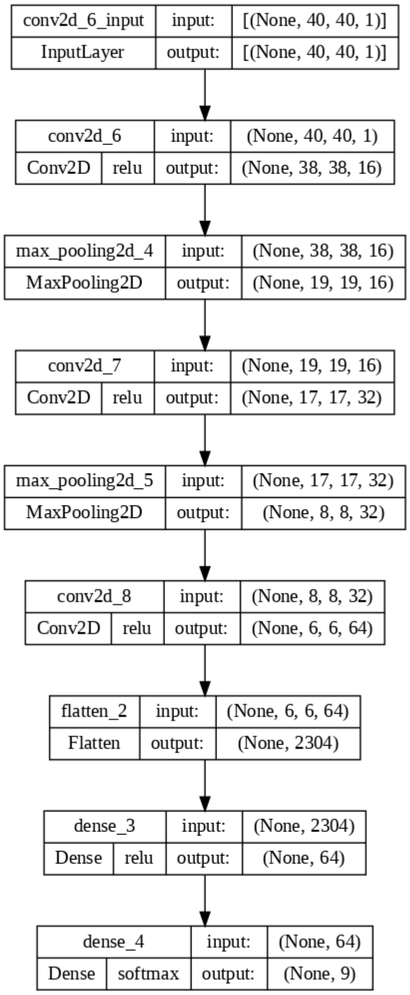


Fig. 3. CNN_Model_1 Architecture

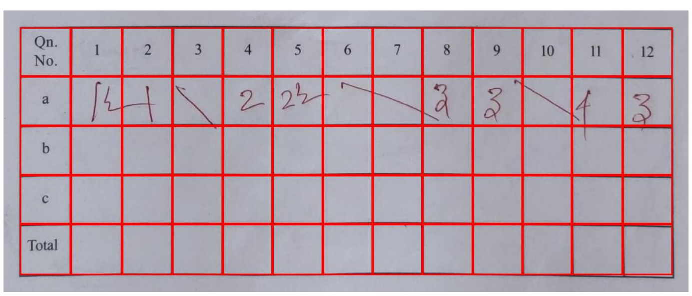


Fig. 4. Sample Output Of Table Detection Process

the resulting DataFrame undergoes post-processing, involving the removal of the first column and both the first and last rows.

Method 2: Cell Coordinates Extraction

This is the method that is used in the work, it primarily focuses on extracting only the cell coordinates using the img2table library and storing them in an ordered dictionary, which maintains the order of key-value pairs. The keys of the dictionary represent rows of mark table, and deleting a key is equivalent to removing a row. After processing, the table cells are cropped using the coordinates from the img2table library and fed into a CNN model.

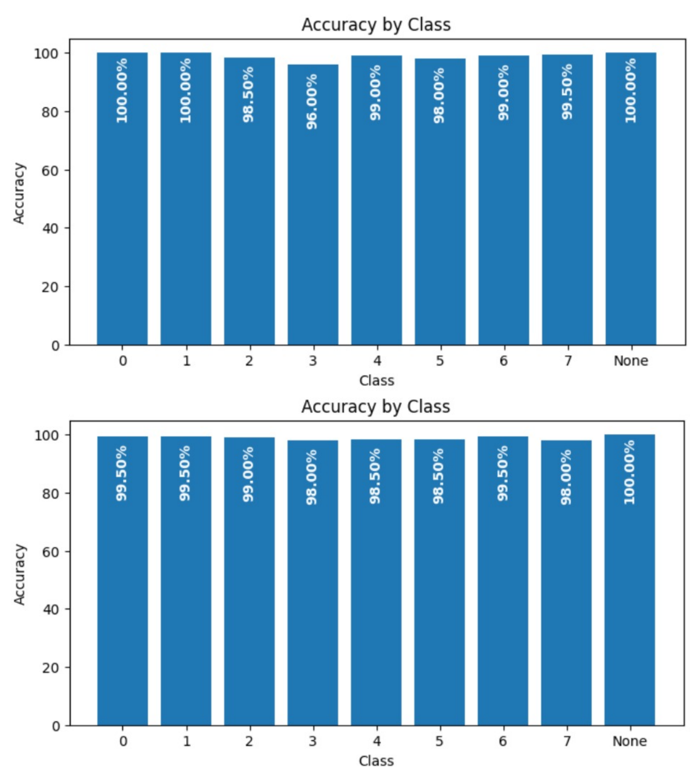
This model can process five tables in just 13 seconds. The output is a DataFrame, that is post-processed to remove the first column. The DataFrame is then converted to a NumPy array to flatten it column-wise, where the matrix-like image data will be converted to a one-dimensional vector for the ease of adding those marks to the main dictionary. . Further post-processing includes removing columns with identical entries and adding Roll No. and Name columns. Finally, the DataFrame is converted to a CSV file, resulting in a refined CSV with student marks.

C. Results And Discussions

The aim of the system is to provide users with a reliable and efficient solution for data entry in mark entry systems that in turn, optimizes productivity and provides more opportunities for the administrative tasks.

To overcome the limitation of first CNN model (CNN_Model_0), a modified model of CNN_Model_0 (CNN_Model_1) was introduced. Both models has achieved a testing accuracy of 99% and 99.2% respectively. For an in-depth comparison, Table 4.3 is useful as it can be seen that class 3 of CNN_Model_0 has a little dip while in the figure of CNN_Model_1, the model has learned all classes equally.

Fig. 5. Accuracy By Class Comparison CNN_Model_0 and CNN_Model_1



Comparing the confusion matrices (given in Table 4.3), CNN_Model_0 showed a slight dip in accuracy for classes 3 and 5, indicating room for improvement. In contrast,

Roll No	Name	1a	2a	3a	4a	5a	6a	7a	8a	9a	10a	11a	12a
00	xx	0	0	3	0	0	4	0	0	0	0	0	0
00	xx	3	0	3	0	0	4	4	0	0	0	0	0
00	xx	3	3	2	0	0	5	0	0	3	0	3	2
00	xx	3	3	2	3	3	0	5	5	3	0	0	7
00	xx	3	3	3	0	0	0	0	0	5	5	0	0

Table I. Output CSV File

CNN_Model_1 demonstrated balanced learning across all classes, indicating its ability to classify instances accurately. These insights gives guidance in refining the models for enhanced performance and accuracy.

The data, given in Table 4.5, compare the performance metrics for the two versions of CNN models. It can be seen that the precision values have not changed with the change in versions. But the recall values have decreased by 0.001 to reach 0.986 in CNN_ Model_1. Also, the F1-scores have decreased by 0.011 to reach 0.988 in CNN_ Model_1.

D. Comparison with the State-of-the-Art Method - Lenet5 vs CNN Model

The Marks2CSV application utilizes the CNN-Model-1, which, while sharing architectural similarities with the renowned LeNet5 model, is tailored for distinct purposes.

CNN-Model-1 exhibits remarkable accuracy, precision, and recall, surpassing the LeNet5 model, especially when optimized with the ADAM optimizer. While CNN-Model-1 excels in a wide range of mark-related tasks, LeNet5 is specialized for detecting numerical and mathematical operations within images [8]. Furthermore, CNN-Model-1 showcases exceptional efficiency by delivering rapid results, enabling efficient data processing and analysis. This efficiency proves advantageous for educational institutions seeking to streamline the mark entry process, making it a valuable tool for enhancing productivity and reducing manual labor in tasks related to image classification and data extraction.

E. Summary

The Marks2CSV application is an asset for mark processing in educational institutions, the system is made using cutting-edge AI technologies like Convolutional Neural Networks and Optical Character Recognition. The profound importance was given to accuracy, precision, and speed. Which makes it capable of processing 5 tables in just 13 seconds. This rapid processing translates to improved productivity for educational institutions, enabling swift access and analysis of marks data, facilitating informed decision-making, and an efficient solution for efficient marks data management.

Fig. 6. Confusion Matrix Comparison CNN_Model_0 and CNN_Model_1

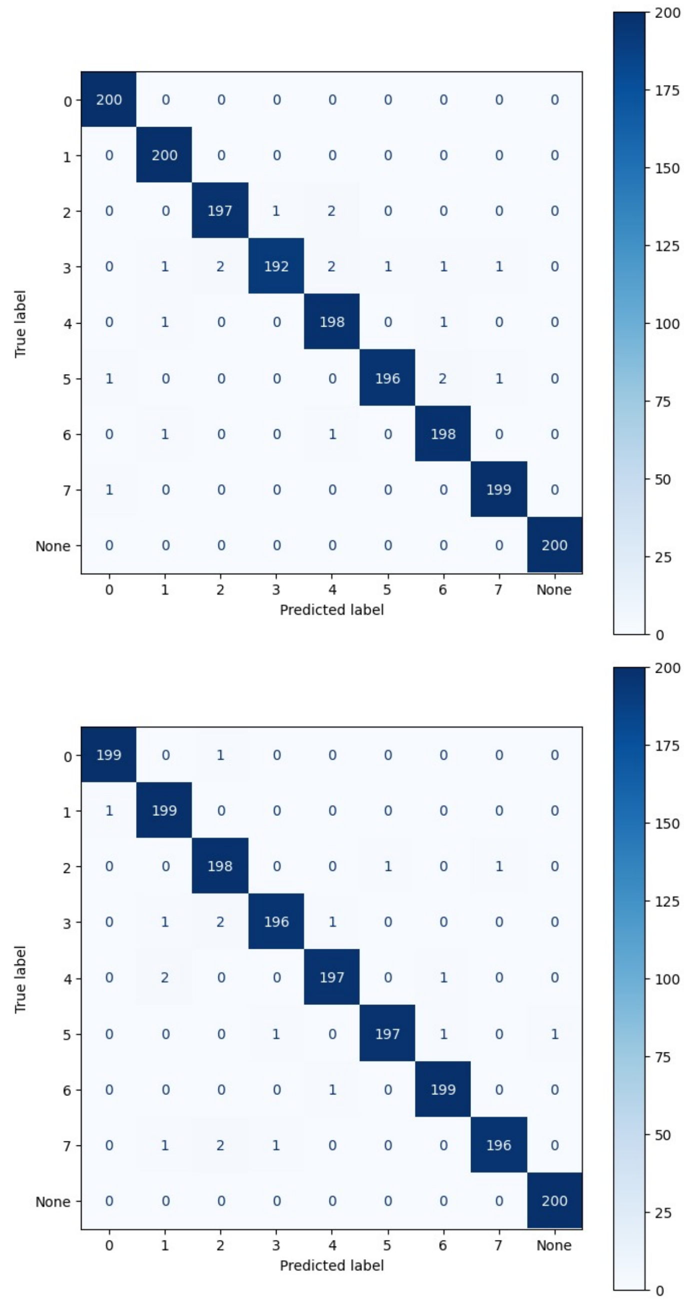


Table II. Performance metrics comparison for two versions of CNN Models

	0	1	2	3	4	5	6	7	None	Overall
CNN_Model_0										
Precision	0.99	0.99	0.99	0.99	0.98	0.99	0.98	0.99	1	0.988
Recall	1	1	0.98	0.96	0.99	0.98	0.99	0.99	1	0.987
F1-Score	1	0.99	0.99	0.98	0.98	0.99	0.99	0.99	1	0.999
CNN_Model_1										
Precision	0.99	0.98	0.98	0.99	0.99	0.99	0.99	0.99	1	0.988
Recall	0.99	0.99	0.99	0.98	0.98	0.98	0.99	0.98	1	0.986
F1-Score	0.99	0.99	0.98	0.98	0.99	0.99	0.99	0.99	1	0.988

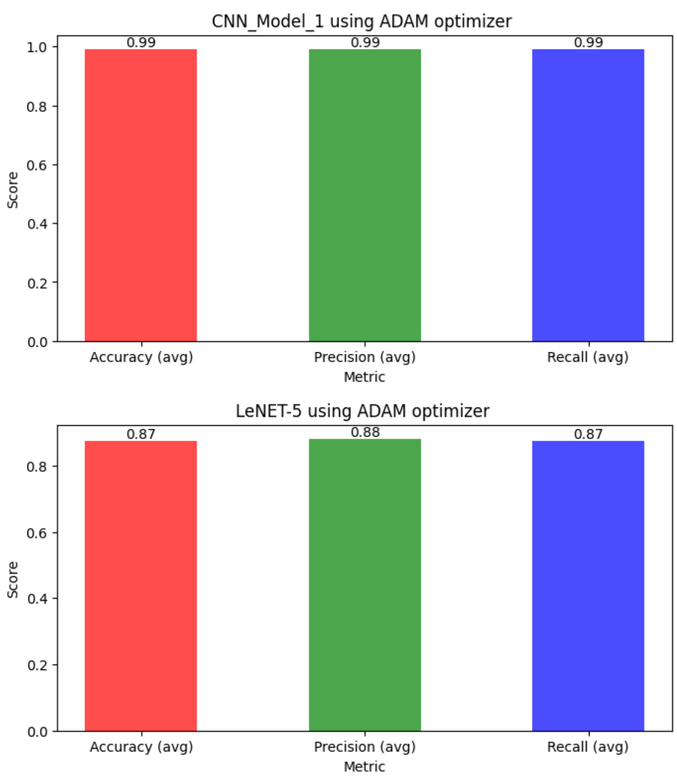


Fig. 7. Model Comparison: CNN Model 1 vs LeNet5 with ADAM Optimizer

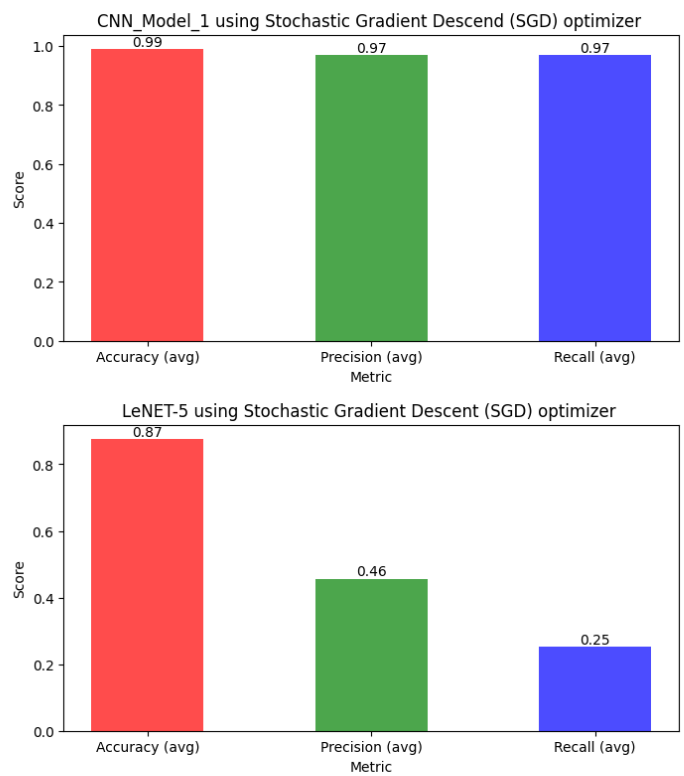


Fig. 8. Model Comparison: CNN Model 1 vs LeNet5 with SGD Optimizer

V. Conclusion

The Marks2CSV application has emerged as an invaluable solution for educational institutions in efficiently converting and processing the marks of students. By delivering outputs in the form of machine-readable and writable CSV files within a matter of seconds, the application significantly accelerates data processing workflows. This expedited turnaround time translates into enhanced productivity, allowing educational institutions to swiftly access and analyze marks data, driving informed decision-making and timely interventions when necessary. However, it is important to acknowledge limitations such as detecting decimal numbers, the efficiency of the OCR tool used, scalability and compatibility with different formats.

VI. References

[1] A. Raj, S. Sharma, J. Singh, A. Singh (Feb 2023), Revolutionizing Data Entry: An In-Depth Study of Optical Character Recognition Technology and Its Future Potential, International Journal for Research in Applied Science & Engineering Technology, Vol. 11 No. 2, pp: 645-653.

[2] Md. Ajij, S. Pratihar, Diptendu S. Roy, T. Hanne (Feb 2022), Robust Detection of Tables in Documents Using Scores from Table Cell Cores, SpringerNature Computer Science Journal, Vol. 3, No. 161, pp: 1-19.

[3] A. Arivoli, D. Golwala, R. Reddy (Oct 2022), CoviExpert: COVID-19 detection from chest X-ray using CNN, Journal of Measurement: Sensors 23, pp: 1-8.

[4] C. ShanWei, S. LiWang, Ng T. Foo, Dzati A. Ramli (Oct 2021), A CNN based Handwritten Numeral Recognition Model for Four Arithmetic Operations, 25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, Procedia Computer Science, Vol. 192, pp: 4416-4424.

[5] Ömer Aydin (Jun 2021), Classification of Documents Extracted from Images with Optical Character Recognition Methods, Anatolian Journal of Computer Sciences, Vol. 6 No. 2 pp: 46-55.

[6] S. Shrivatsava, Sanjeev K. Singh, K. Shrivatsava, V. Sharma (Mar 2021), CNN-based Automated Vehicle Registration Number Plate Recognition System 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), IEEE Xplore, pp: 795-802.

[7] Colin G. White-Dzuro, Jacob D. Schultz, C. Ye, Joseph R. Coco, Janet M. Myers, C. Shackelford, S. T. Rosenbloom, D. Fabbri (Mar 2021), Extracting Medical Information from Paper COVID-19 Assessment Forms, Applied Clinical Informatics Vol. 12 No. 1, pp: 170–178.

[8] Md. R. Islam, A. Matin (Dec 2020), Detection of COVID 19 from CT Image by The Novel LeNet-5 CNN Architecture, 23rd International Conference on Computer and Information Technology (ICCIT), pp: 1-5.

[9] J. Li, G. Sun, L. Yi, Q. Cao, F. Liang, Y. Sun (Nov 2020), Handwritten Digit Recognition System Based on Convolutional Neural Network, IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), pp: 739-742

[10] J. Yuan, H. Li, M. Wang, R. Liu, C. Li, B. Wang (Sep 2020), An OpenCV-based Framework for Table Information Extraction, 2020 IEEE International Conference on Knowledge Graph (ICKG), IEEE Xplore, pp: 621-628.

[11] A. Das, Gyana R. Patra, Mihir N. Mohanty (Sep 2020), LSTM based Odia Handwritten Numeral Recognition, 2020 International Conference on Communication and Signal Processing (ICCSP), IEEE Xplore, pp: 538-541.

[12] J. Memon, M. Sami, Rizwan A. Khan, M. Uddin (Aug 2020), Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR), IEEE Access, Vol. 8, pp: 142642-142668.

[13] P. Jawale, Hitiksha P. Chaudhary, N. Rajput (Aug 2020), Real-Time Object Detection using TensorFlow, International Research Journal of Engineering and Technology (IRJET), Vol. 7, Issue 8, pp: 5015-5019.

[14] A. Yaganteeswarudu (July 2020), Multi Disease Prediction Model by using Machine Learning and Flask API 2020, 5th International Conference on Communication and Electronics Systems (ICCES), IEEE Xplore, pp: 1242-1246.

[15] Raajkumar G., Indumathi D. (July 2020), Optical Character Recognition using Deep Neural Network, International Journal of Computer Applications, Vol. 176 No. 41, pp: 61-65.

[16] Vaisakh V K, Lyla B Das (June 2020), Handwritten Malayalam Character Recognition System using Artificial Neural Networks, IEEE International Students' Conference on Electrical, Electronics and Computer Science, pp: 1-4.

[17] B. Barz, J. Denzler (May 2020), Deep Learning on Small Datasets without Pre-Training using Cosine Loss, 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE Xplore, pp: 1360-1369.

[18] T.V. Janahiraman, M.S.M. Subuhan (Oct 2019), Traffic Light Detection Using Tensorflow Object Detection Framework, IEEE 9th International Conference on System Engineering and Technology (ICSET), pp: 108-113.

[19] Teddy S. Gunawan, A.F.R. Mohd. Noor, M. Kartiwi (May 2018), Development of English Handwritten Recognition Using Deep Neural Network", Indonesian Journal of Electrical Engineering and Computer Science, Vol. 10, No. 2, pp: 562-568

[20] B. Gatos, D. Danatsas, I. Pratikakis, S. J. Perantonis (2005), Automatic Table Detection in Document Images, International Conference on Pattern Recognition and Image Analysis (ICAPR), Pattern Recognition and Data Mining, pp: 609–618.