Handwritten Text Recognition System using Machine Learning

[1]Dr. D. Saraswathi, [2]Sanaa Mohamed Sherif

[1]*Associate Professor, Department of Computer Science, PSG College of Arts & Science, Coimbatore-641 014.*

[2]*Department of Computer Science, PSG College of Arts & Science, Coimbatore-641 014.*

[1]*saraswathi.ds@gmail.com* [2]*sanaasheriff1234@gmail.com*

Abstract

Handwritten character recognition is an ongoing research field that features machine learning, computer vision and pattern recognition. To do this, one scans a handwritten document and converts it into a simple text document. The basic Optical Character Recognition (OCR) process is to examine the text of a document and convert it into codes used for data processing. In this machine learning project, deep learning techniques were used to model a neural network that recognizes individual handwritten characters and handwritten numerals. To recognize them, a convolutional neural network (CNN) was built to train on alphabets and the digits datasets and further the predictions done by the trained model were visualized using OpenCV.

Keywords: *handwritten character recognition, ocr, machine learning, handwritten digits recognition, cnn, deep learning*

1. Introduction

Over the past two decades, advances in handwritten word processing have enabled us to increase the performance of systems dedicated to transcribing, indexing, and semantic interpretation of handwritten text documents. Handwriting recognition (HWR, also recognized as handwritten text recognition or HTR) is the potential of a computer to read and interpret intelligible handwritten input from a source such as paper documents, photos, touchscreens and, other devices.[18] Automatic text recognition of handwriting is a complicated problem that requires a careful combination of several advanced pattern recognition techniques. The computer that recognizes the handwriting should be able to capture, recognize, and convert characters from paper documents, images, touchscreen devices, and other sources into machine-coded form.

The use of neural networks as a database rather than with classical methods means no manual hardcoding is required. Instead, the parameters are learnt during the training process. This makes deep learning methods more resilient to changes in handwriting styles, and reduces the challenges in feature extraction in classical methods. However, the output accuracy depends on the quality and completeness of the data set used in the training process.[20] Convolutional neural networks are Deep Learning algorithms that are widely utilized in computer visualization and have become state-of-the-art in numerous visual applications such as image classification and have also succeeded in processing natural language, text classification. The use of convolutional layers in networks reflects the structure of the human visual cortex, where a number of layers process incoming images to identify complex features. In this project, a neural network model was built and trained over the datasets consisting of images of alphabets and digits.

1.1. Open CV

The Open Source Computer Vision Library also mentioned as OpenCV is a library of programming features aimed at Real-Time Computer Vision. Computer vision is the process that enables everyone to understand images and videos, how they are stored and how they can be

manipulated to retrieve the data from them. The OpenCV project, launched in 1999, is an Intel research initiative to promote CPU-intensive applications and is part of a series of projects, including beam tracking and 3D wall displays.[17] In this project, OpenCV was used to view the predicted output.

2. Literature Review

In this section, the literature survey that was made to recognize the various image pre-processing algorithms and neural networks for building the model has been discussed, and conclusions were drawn based on the respective survey.

2.1. Review on Image pre – processing

Image pre-processing is a crucial step to filter unnecessary features that may degrade the performance of the model. It is a process where steps are implemented to format images before they're used for modelling, and training purposes. Examples such as resizing images, converting images to grayscale, and increasing the dimensions, etc.

Table 1 survey on image pre-processing

Reference number	*Paper(s)*	*Descriptions*
[1]	Handwritten Character Recognition in English: A Survey [Monica Patel et al, 2015]	Binarization, noise removal and skew correction was done.
[2]	A Review Paper on Handwritten Character Recognition Using Machine Learning [Bhagyashree D. Upadhyay et al 2020]	applied spatial image filtering, global image thresholding process to enhance the images.
[3]	Handwritten Character Recognition Using Cnn [S. Anandh Kishan et al, 2018]	Resizing, converting images to grey-scale.
[4]	Diagonal Based Feature Extraction For Handwritten Alphabets Recognition System Using Neural Network [J.Pradeep et al, 2011]	Noise removal, Binarization, Edge detection, Dilation and Filling were performed to pre–process the images.
[5]	Preprocessing Techniques for Online Handwriting Recognition [Bing Quan Huang et al, 2009]	removing the hooks of the strokes by applying a changed angle threshold with a length threshold.
[6]	Preprocessing Techniques in Character Recognition [Y. Alginahi, 2010]	Spatial image filtering operations were explained
[7]	Handwritten character recognition using convolutional neural network [I Khandokar et al, 2020]	Enhanced images by removing noise, converting to grey scale and by performing binarization.
[8]	Full Page Handwriting Recognition via Image to Sequence Extraction [Sumeet S. Singh et al, 2021]	Data augmentation such as Image scale, rotation, brightness, background colour of synthetic images, contrast, and Gaussian noise were performed.
[9]	Handwritten Character Recognition Using BP NN, LAMSTAR NN and SVM [Majed Valad Beigi, 2015]	Morphology at step image dilation and image filling were performed.
[10]	How to Make Real-Time Handwritten Text Recognition With Augmentation and Deep Learning [Sushant Gautam, 2020]	Data Augmentation technique to increase the accuracy and ability to work in real-time

From the survey, it was understood that various image pre-processing techniques like noise removal, binarization, normalization, modifying certain image details to maximize the quality for

computer vision, etc., were mostly performed and how it plays a vital role in helping to overcome problems that occur due to low-quality images and thus giving better image recognition.

2.2. Review on Neural Networks

Among various deep learning neural architectures, Convolutional neural networks also known as CNN has been the most impressive out of all the other algorithms, mainly due to their ability to recognize different unique patterns from the image on their own. A CNN is a deep learning neural network built for processing structured arrays of data like images.

Table 2 survey on neural networks

Reference number	***Paper(s)***	***Descriptions***
[9]	Handwritten Character Recognition Using BP NN, LAMSTAR NN and SVM [Majed Valad Beigi et al, 2015]	SVM and neural networks have been used for feature extraction
[10]	How to Make Real-Time Handwritten Text Recognition With Augmentation and Deep Learning [Sushant Gautam et al, 2020]	CNN for performing classification.
[11]	A survey: Comparison between Convolutional Neural Network and YOLO in image identification [Richeng Cheng et al, 2020]	Have used R-CNN to solve detection in object classification problems and bounding boxes.
[12]	The use of machine learning algorithms for image recognition [Jan Matuszewski et al, 2020]	To recognize objects in images, an artificial neural network was used (ANN).
[13]	Embedded real-time speed limit sign recognition using image processing and machine learning techniques [Gomes, S.L. et al, 2017]	Proposed the detection and recognition of speed limit signs using SVM, OPF and KNN classifiers.
[14]	A very high accuracy handwritten character recognition system for Farsi/Arabic digits using Convolutional Neural Networks [S. S. Ahranjany et al, 2010]	Automatic extraction of input pattern's features using CNN.
[15]	Persian handwritten character recognition using convolutional neural network [B. Alizadehashraf et al, 2017]	SCNN and ECNN methods were performed.
[16]	Deep learning-based large scale handwritten Devanagari character recognition [S. Acharya et al, 2015]	Have used Deep CNN to increase accuracy.
[19]	A Study of Image Pre-processing for Faster Object Recognition [Md Tanzil Shahriar et al, 2020]	Have used various algorithms such as SVM, R-CNN, YOLO.

From the survey, it was seen that mostly neural network algorithms such as R-CNN, CNN, ANN, ECNN, SCNN etc were used and thus in conclusion mainly convolutional neural networks were used for optical character recognition and text classification in general.

3. Research Contribution

The proposed work consists of five phases for handwritten text recognition. The phases are data collection, data pre-processing, neural network creation, model training and prediction. The flow of mechanism is depicted for the proposed study in Figure 1.

Proposed Algorithm

1. Dataset Collection
2. Image Acquisition
3. Image pre-processing
4. Convolutional Neural Network
5. Classification
6. Prediction using Open CV

At first, the datasets were collected and read in their binary form. Further, the data was split into images and their corresponding labels and so a column was dropped from the data frame and was used as a variable to form the labels. Then, the training and testing images were pre-processed i.e., they were resized or reshaped, then were converted to grey scale images, normalization was performed for even distribution of the data, dimensions were given such that the neural network model can extract the unique features from the images more precisely, and a dictionary was created for alphabets to map the values with the data. To describe the distribution of the data, a horizontal bar plot was drawn as shown in Figure 2. The pre-processed data was then passed as input to the neural network model for feature extraction and finally compiling and fitting of the model was done.

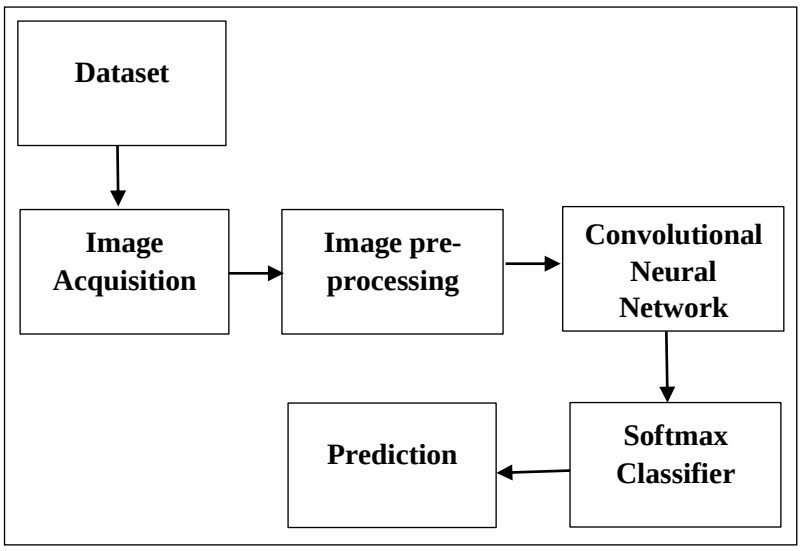

Figure 1 proposed architecture

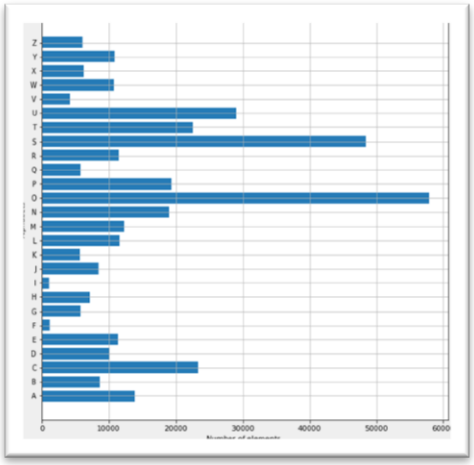

Figure 2 amount of data in character dataset

4. Normalization

The purpose of data normalization is to reduce bias in the model training process by balancing the inputs and outputs of a variable. This is done by scaling the input variables so that they have a mean of zero and a variance of one. Data can be normalized using several methods such as dividing each value by its standard deviation, subtracting the mean value from each variable, or multiplying each value with a factor that scales its range to be between -1 and 1. The images have been normalized for the even distribution of the data such that the accuracy can be improved.

5. Neural Network

The proposed neural network for this project is a Convolutional neural network also known as CNN. Convolutional neural networks comprise an input layer, a hidden layer and an output layer. In upstream neural networks, the middle layers are called hidden because their inputs and outputs are disguised by the activation function of the final folding. Convolutional pool layers tend to use the relu function, while FC layers use the softmax activation function to classify inputs and generate probabilities of 0 or 1. An example of the working of a CNN has been depicted in Figure 3.

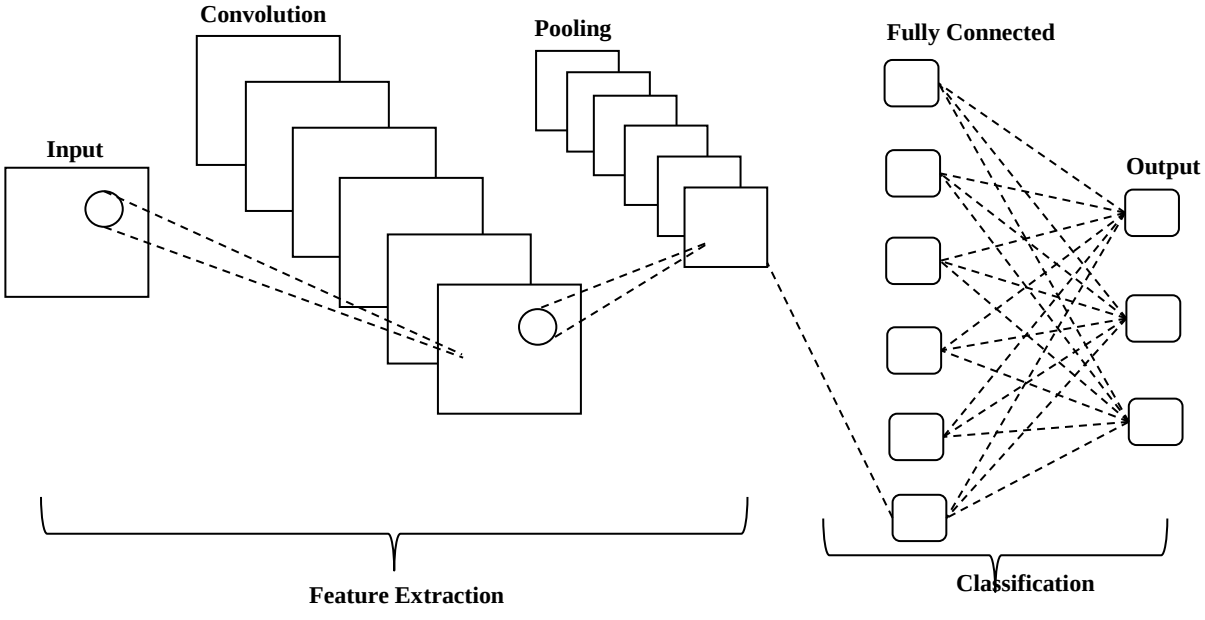

Figure 3 Convolutional Neural Network

5.1. Layers

In deep learning, neurons are organized into several layers. A neuron in a layer is connected to all neurons in the preceding and following layers. They can be combined and grouped so that the neurons in a layer are connected to a single neuron in the next layer, decreasing the number of neurons in that layer. Each layer learns features by locating regions of the scanned image. Convolutional layers like Convo2D, max pool, Dense were used to perform the feature extraction in the neural network.

5.1.1. Convolutional Layer

A Convolutional layer is formed from neurons connected to the sub-regions of the input image or the output of the previous layer. Each layer learns features by locating regions of the scanned image. To create a layer with the Convo2D layer function, the size of each region with the argument filter size was specified. Multiple filters or kernels have been used to run over the image giving a dot product. Each filter has extracted different features from the image. The feature had to be of the same size as the input image, so padding was done which is a technique that adds zeroes around the margin of the image to increase its dimension.

5.1.2. Pooling

When one coil layer is added to the model, Max pooling reduces the dimension of the image by reducing the amount of pixels output from the previous coil layer. This layer helps in reducing the spatial size of the convoluted features and the over-fitting by providing an abstract representation of them. The convolutional layers were followed by max pool layers that were used for reducing size of the images.

5.1.3. Activation Function (ReLU)

The convolutional layer applies the ReLU activation function to the output and introduces non-linearities into the model. ReLU function is a piecewise linear function that outputs the input instantly if it's positive i.e., greater than zero, if not, it'll output zero.

5.1.4. Dense Layer

By default, the dense layer is a single neuron output layer with a linear activation function. The basic parameter is the parameter it uses, which is a positive integer, and its value represents the output size of the plane. Ultimately the output of the max pool layers and convolution layers were flattened into a vector of single dimension using the Flatten() function and was given as the input to the Dense layer.

5.1.5. Softmax classifier

Softmax activation functions are mostly used in neural networks when it is necessary to build a multi-class classifier that solves the problem of assigning a single instance of a class to any number of possible classes bigger than two. The softmax function works on the output layer node by printing a value for each node and the probability that the sum of these values will result in 1.

The model summary describes what were the different layers defined in the character model and digits model as shown in Table 3 and Table 4.

Table 2 model summary of character recognition

Layer (type)	***Output Shape***	***Param***
max_pooling2d (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_1 (Conv2D)	(None, 13, 13, 64)	18496
max_pooling2d_1 (MaxPooling2	(None, 6, 6, 64)	0
conv2d_2 (Conv2D)	(None, 4, 4, 128)	73856
max_pooling2d_2 (MaxPooling2)	(None, 2, 2, 128)	0
flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 64)	32832
dense_1 (Dense)	(None, 128)	8320
dense_2 (Dense)	(None, 26)	3354

Total params: 137,178

Trainable params: 137,178

Non-trainable params: 0

Table 3 model summary for digits recognition

Layer (type)	***Output Shape***	***Param***
conv2d (Conv2D)	(None, 26, 26, 64)	640
activation (Activation)	(None, 26, 26, 64)	0
max_pooling2d (MaxPooling2D)	(None, 13, 13, 64)	0
conv2d_1 (Conv2D)	(None, 11, 11, 64)	36928
activation_1 (Activation)	(None, 11, 11, 64)	0
max_pooling2d_1 (MaxPooling2)	(None, 5, 5, 64)	0

conv2d_2 (Conv2D)	(None, 3, 3, 64)	36928
activation_2 (Activation)	(None, 3, 3, 64)	0
max_pooling2d_2 (MaxPooling2)	(None, 1, 1, 64)	0
flatten (Flatten)	(None, 64)	0
dense (Dense)	(None, 64)	4160
activation_3 (Activation)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2000
activation_4 (Activation)	(None, 32)	0
dense_2 (Dense)	(None, 10)	330

Total params: 81,066

Trainable params: 81,066

Non-trainable params: 0

Fitting Model

Here, the compiling, defining the optimizing function and the loss function to be used for fitting the model has been discussed.

5.2. Optimization

An optimizer was used to solve the optimization problem by minimizing a function. In the case of machine learning, optimization refers to the procedure of minimizing the loss function and updating network weights. The optimizing function used here was Adam, which is a combination of RMSprop & Adagram optimizing algorithms.

5.3. Loss

A loss function works with errors and quantifies how bad it is that the error of a certain size and direction affects the negative consequences of an incorrect forecast. The cross-entropy loss function is an optimization function used to classify data when training a classification model by predicting the likelihood that the data belong to a class or another class.

6. Training and Datasets

The defined CNN model was trained on the train images. 70% percent of the training data was used for training the model and 30% for validation i.e., for validating itself and since the volume of the data was huge it was trained only for a single epoch, however, it can be trained for multiple epochs which is recommended for better recognition and accuracy. epoch refers to the number of passes of the entire training dataset the algorithm has completed.

The Dataset for characters (NIST and NMIST) was collected from Kaggle.com datasets [21]. This dataset consists of around 370000+ images. The dataset contains 26 folders (A-Z) containing handwritten images in size 28x28 pixels, each alphabet in the image is centre fitted to a 20x20 pixel box. And MNIST dataset for digits was collected from Kaggle.com [22].

7. Experimental Results

When the model was trained and saved. The training and validation accuracies along with the training and validation losses for character recognition and digits recognition were observed as shown in Table 5 and Table 6.

Table 5 accuracy and loss for characters

	Loss	***Accuracy***
Validation	0.0801	0.9781
Training	0.1548	0.9575

Table 6 accuracy and loss for numerals

	Loss	***Accuracy***
Validation	0.1044	0.9781
Training	0.0978	0.9761

Finally, 9 subplots in the shape (3,3) was created to visualize few test data alphabets along with their predictions, that was made by the model for character recognition as shown in Figure 4.

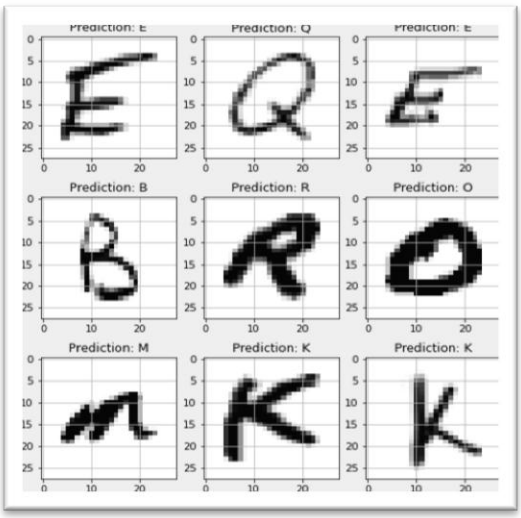

Figure 4 prediction on test data

7.1. Prediction on External Image

Predictions on external images were done using OpenCV for character recognition and live predictions for digits recognition were done by giving the input via Microsoft Paint and screenshots were captured using a python library known as pyscreenshot as shown in Figure 5 and Figure 6.

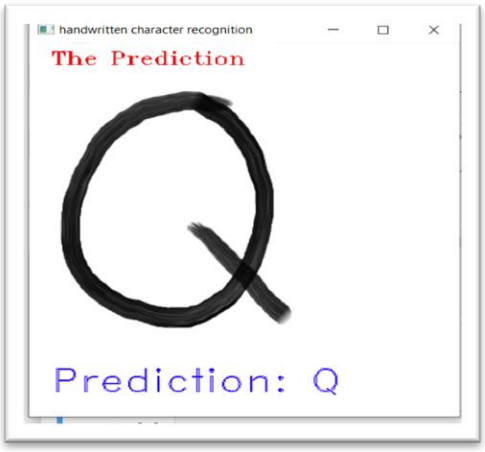

Figure 5 prediction on external character image

Figure 6 live prediction on digits

8. Limitations

The variety of human fonts, and irregularities in the handwriting can lead to less precise character recognition. When the letters are connected, it would become more difficult for the computer to do the prediction hence, the model is limited to recognise only individual characters and digits.

9. Conclusion

Neural network models based on handwritten characters and digits were introduced in this paper for classifying and recognizing alphabets and numerals. The pixel values derived from resized characters and numerals were directly used for training the neural network. Of the several neural networks used for classifying the characters and numerals, these models yield the highest accuracies of 97.65% and 97.91%.

10. Future Enhancement

A lot of research would still be required for improving the current performance. Certain methods can be used for increasing better recognition rate. It can be extended to recognize words, sentences and paragraphs with the help of the segmentation process. The model could be developed for recognizing different languages.

11. Acknowledgements

I convey my heartiest and passionate sense of thankfulness to my guide Dr. D. Saraswathi M.Sc., M.Phil., Ph.D., Associate Professor, Department of Computer Science (PG – Software Systems), for her/his timely suggestion which had enabled me in completing the project successfully.

This note of acknowledgement will be incomplete without paying my heartful devotion to my parents, my friends and other people, for their support, encouragement, and the patience, without which it would have been impossible for me to complete the job.

References

[1] Patel, Monica & Thakkar, Shital. (2015). Handwritten Character Recognition in English: A Survey. IJARCCE. 4. 345-350. 10.17148/IJARCCE.2015.4278.

[2] Bhagyashree D. Upadhyay, Dr. Sachin Chaudhary, Monali Gulhane, "A Review Paper on Handwritten Character Recognition Using Machine Learning", International Journal of Future Generation Communication and Networking Vol. 13, No. 2s, (2020), pp. 1256–1266, ISSN: 2233-7857 IJFGCN.

[3] S. Anandh Kishan, J. Clinton David, P. Sharon Femi, Sri Venkateswara, "Handwritten Character Recognition Using Cnn", IJRAR September 2018, Volume 5, Issue 3, E-ISSN 2348-1269, P- ISSN 2349-5138.

[4] J. Pradeep, E. Srinivasan, S. Himavathi, "Diagonal Based Feature Extraction For Handwritten Alphabets Recognition System Using Neural Network", International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 1, Feb 2011.

[5] Bing Quan Huang, Y. B. Zhang, M. -T. Kechadi et al, "Preprocessing Techniques for Online Handwriting Recognition", In Nedjah N., de Macedo Mourelle L., Kacprzyk J., França F.M.G., de De Souza A.F. (eds) Intelligent Text Categorization and Clustering. Studies in Computational Intelligence, vol 164. Springer, Berlin, Heidelberg.

[6] Y. Alginahi, "Preprocessing Techniques in Character Recognition", Semantic Scholar, 2010.

[7] I Khandokar, Md M Hasan, F Ernawan, Md S Islam, M N Kabir, "Handwritten character recognition using convolutional neural network", 2021, J. Phys.: Conf. Ser. 1918 042152.

[8] Sumeet S. Singh, d Sergey Karaye, "Full Page Handwriting Recognition via Image to Sequence Extraction", arXiv:2103.06450v2 [cs.CV] 21 May 2021.

[9] Majed Valad Beigi, "Handwritten Character Recognition Using BP NN, LAMSTAR NN and SVM", EECS 349, Northwestern University, 2015.

[10] Sushant Gautam, "How to Make Real-Time Handwritten Text Recognition With Augmentation and Deep Learning", Medium, Dec 15, 2020.

[11] Richeng Cheng, "A survey: Comparison between Convolutional Neural Network and YOLO in image identification", Journal of Physics: Conference Series 1453. 012139, 2020.

[12] Jan Matuszewski, Adam Rajkowski, "The use of machine learning algorithms for image recognition", Proceedings Volume 11442, Radioelectronic Systems Conference 2019, 1144218 (2020).

[13] Gomes, S.L., Rebouças, E.d.S., Neto, E.C. et al. Embedded real-time speed limit sign recognition using image processing and machine learning techniques. Neural Comput & Applic 28, 573–584 (2017).

[14] S. S. Ahranjany, F. Razzazi and M. H. Ghassemian, "A very high accuracy handwritten character recognition system for Farsi/Arabic digits using Convolutional Neural Networks," 2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), 2010, pp. 1585-1592.

[15] B. Alizadehashraf and S. Roohi, "Persian handwritten character recognition using convolutional neural network," 2017 10th Iranian Conference on Machine Vision and Image Processing (MVIP), 2017, pp. 247-251.

[16] 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), 2015, pp. 1-6.
[17] "OpenCV" Wikipedia, Wikimedia Foundation, 7 August 2021.
[18] "Handwriting recognition" Wikipedia, Wikimedia Foundation, 3 October 2021.
[19] Md Tanzil Shahriar, Huyue Li, (2020), "A Study of Image Pre-processing for Faster Object Recognition".
[20] Rachel Wiles, "Have we solved the problem of handwriting recognition?", Towards Data Science, June 27, 2019.
[21] Sachin Patel, "A-Z Handwritten Alphabets in .csv format", 370000+ English Alphabets Image Data-set, Kaggle.
[22] Dariel Dato-on, Joseph Redmon, "MNIST in CSV", Kaggle.