Recognition and Digitization of Handwritten Text using Histogram of Gradients and Artificial Neural Network

Dr.S.K.Nivetha, Dr.M.Geetha, Dr.R.S.Latha, C.Vignesh, P.Vijay, N.Vijaya Kumar
Department of CSE, Kongu Engineering College, Perundurai, Erode, TN, India
nivethasen@gmail.com

Abstract: Handwriting recognition is one of the most persuasive and interesting projects as it is required in many real-life applications such as bank-check processing, postal-code recognition, handwritten notes or question paper digitization etc. Machine learning and deep learning methods are being used by developers to make computers more intelligent. A person learns how to execute a task by learning and repeating it over and over before it memorises the steps. The neurons in his brain will then be able to easily execute the task that he has mastered. This is also very close to machine learning. It employs a variety of architectures to solve various problems. Handwritten text recognition systems are models that capture and interpret handwritten numeric and character data from sources such as paper documents and photographs. For this application, a variety of machine learning algorithms were used. However, several limitations have been found, such as a large number of iterations, high training costs, and so on. Even though the other models have given impressive accuracy, it still has some drawbacks. In an unsupervised way, the Artificial Neural Network is used to learn effective data coding. For recognising real-world data, we built a model using Histogram of Oriented Gradients (HOG) and Artificial Neural Networks (ANN).

Keywords: Handwritten text recognition, Histogram of Gradients, Artificial Neural Network, Digitization.

1. Introduction

Since everybody in the world has their own writing style, handwriting recognition is one of the most persuasive and interesting projects. Handwritten recognition systems are models that gather and interpret handwritten numeric or character information from sources such as paper documents and photographs. It is the computer's ability to automatically recognise and perceive handwritten digits or characters. This recognition is needed in real-world applications such as the translation of handwritten information into digital format, number plate recognition, bank check processing, postal code recognition and signature verification.

Handwriting recognition refers to a computer's ability to receive and interpret intelligible handwritten information from a variety of sources, including paper, photographs, touch screens and other devices. Handwriting recognition, also called handwritten text recognition, is still a difficult problem to solve (Manchala *et al* 2020). Converting handwritten text to machine readable text is difficult due to the wide range of handwriting styles among people and the low quality of handwritten text compared to printed text. It is a critical issue for various sectors, including insurance, banking and healthcare.

Deep learning is an artificial intelligence branch of machine learning that uses neural networks to learn unsupervised from unstructured or unlabeled data. Deep neural learning, or deep neural network, is another name for it (Ali *et al* 2019). Deep learning is an artificial intelligence function that imitates the processing of data that happens in the human brain to recognise expression, identify objects, make decisions and translate languages. Deep learning is able to learn using unlabeled and structured data without the help of humans. Deep learning is a form of machine learning that can aid in the recognition of handwritten characters and numbers from a variety of sources. Deep Learning depends on the structure of the human brain. Deep learning algorithms analyse data with a predetermined logical framework in order to draw similar conclusions as humans. Deep learning employs a multi-layered approach to accomplish this.

The architecture of the neural network is based on the configuration of the human brain. Neural networks can be trained to perform the same tasks on data as our brains do when identifying patterns and classifying various types of knowledge. When we obtain new knowledge, our brain attempts to equate it to previously encountered objects. Deep neural networks make use of the same principle (Qiao *et al* 2018).

We may use neural networks to perform a variety of tasks, such as clustering, sorting, and regression. We can use neural networks to group or sort unlabeled data based on similarities between the samples. We may train the network on a classified dataset to classify the samples in this dataset into different categories in the case of classification.

A Logistic Regression can be compared to a single perceptron (or neuron). At each layer of the Artificial Neural Network, there are several perceptrons/neurons. Since inputs are only interpreted in one direction, an ANN is also known as a Feed-Forward Neural Network.

Three layers make up ANN. Input, Hidden, and Output are the three categories. The data is received by the input layer, the processing of data is done in the hidden layer, and the outcome is displayed by the output layer. Each layer makes an effort to master specific weights. Problems involving tabular data, image data, and text data can all be solved using ANN. The ANN architecture is represented in Figure.1. In this architecture, there is one input layer, multiple hidden layers and one output layer is there.

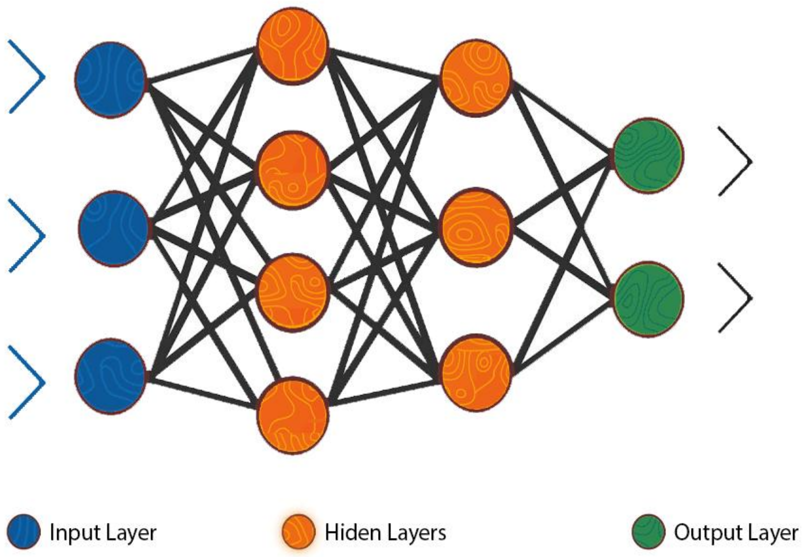


Figure 1. Artificial Neural Network

1.1 Advantages of artificial neural network

Any nonlinear function can be learned by an Artificial Neural Network. As a consequence, Universal Function Approximators is a common name for these networks. ANNs will learn weights that map any input to the desired output. The activation function is a key explanation for universal approximation. The network's nonlinear properties are introduced by activation functions. This aids the network in learning every dynamic input-output relationship. An activation feature is an ANN's powerhouse.

1.2 Feedforward neural network

In a feedforward neural network, the links between each node do not form a cycle. The feedforward neural network was the first artificial neural network to be created, and it was the easiest (Trivedi *et al* 2018). Knowledge only flows in a single direction (forward) in this network, moving through the hidden nodes from the input to the output nodes. Cycles and loops do not exist in the network. When learning data that isn't sequential or time-based, feedforward neural networks are most commonly used.

2. Literature Review

Handwriting recognition systems are models used to obtain and interpret information in the form of handwritten texts from sources like paper records and images. The success of handwriting recognition systems was strongly focused upon the optical text recognition which is responsible for segmentation of handwritten digit and character recognition is the soul of this module (Bora *et al* 2020). The matching of handwritten digits and characters to their accompanying electronic records is done using number and character recognition. Any kind of learning paradigm will do this.

Trivedi *et al* (2018) proposed Hybrid evolutionary approach for Devanagari handwritten numeral recognition using Convolutional Neural Network. They used some techniques to develop the model namely Sparse Autoencoder, CNN, Softmax Classifier, Genetic Algorithm. For CNN training, a hybrid deep learning approach using the Genetic Algorithm and the L-BFGS system has been developed. To test the model, Devanagiri handwritten numeral dataset has been taken. The conclusion of this paper is that evolutionary techniques should be used to more effectively train CNN. If the number of iterations are increased, there will be some problems in length of chromosomes. Genetic algorithms are not applied on each layer of the system.

Shamim et *al* (2018) proposed Handwritten Digit Recognition using Machine Learning Algorithms. Bayes Net, Support Vector Machine, Random Forest, Random Tree, Multilayer Perceptron, J48 and Naive Bayes were the techniques used. This paper describes a method for recognising handwritten digits off-line using various machine learning techniques. The key goal of this paper is to ensure that methods for recognising handwritten digits are both accurate and efficient. WEKA was used to recognise digits using a variety of machine learning algorithms, including Bayes Net, Support Vector Machine, Random Forest, Random Tree, Multilayer Perceptron, J48 and Naive Bayes. The maximum accuracy gained in this paper is 90.37% using Multilayer perceptron. Accuracy needs to be improved further.

SubbaRao Gogulamudi *et al* (2020) proposed Handwritten Digit Recognition by using Pattern Recognition & Consensus Clustering. Pattern Recognition and Consensus Clustering are the methods used in this article. The aim of this paper is to use Pattern Recognition and Consensus Clustering to create a Handwritten Digit Recognition System. The MNIST dataset (Modified National Institute of Standards and Technology database) of Handwritten numeral dataset is used. This implementation is done in the language Python with the Anaconda package and Consensus Clustering package by using the tool Jupyter Notebook. Clustering algorithms can make it difficult to interpret data, particularly when there is no way of knowing how many clusters there are.

Bora *et al* (2020) proposed Handwritten Character Recognition from Images using CNN-ECOC. OCR is presented in this paper using a CNN and an Error Correcting Output Code (ECOC) classifier. The CNN is used to isolate features, and the ECOC is used to classify them. The NIST handwritten character image dataset was used to train and validate the CNN-ECOC. In comparison to the standard CNN classifier, CNN-ECOC has better precision, according to this article. One disadvantage of ECOC classifier is that it imposes a greater computational demand leading to longer training times.

Manchala *et al* (2020) proposed Handwritten text recognition using deep learning with TensorFlow. The techniques used in this paper are Connectionist temporal classification (CTC), Recurrent Neural Network (RNN) and Convolutional Neural network (CNN) that is used for training recurrent neural networks. The model is implemented using TensorFlow. The accuracy obtained by this technique is 90.3%. Only the text with the least amount of noise is given the highest precision in this project. The accuracy completely depends on the dataset. If the data is more, the accuracy obtained by this model is more. It does not give best accuracy for cursive letters.

Hallale & Salunke (2013) proposed Twelve Directional Feature Extraction for Handwritten English Character Recognition. Twelve directional features are used to recognise handwritten English alphabets and numerals in this paper. With directional pattern matching, the properties of similarity measures are investigated. The detection rate of traditional and twelve directional feature extraction techniques is then compared. The experiment demonstrates that directional function extraction techniques outperform traditional methods. But the accuracy achieved by this project is only 88.29%. Accuracy needs to be improved further.

Choudhary *et al* (2013) proposed Off-line Handwritten Character Recognition Using Features Extracted from Binarization Technique. The key goal of this project is to remove features obtained via the binarization method for the identification of handwritten English characters. To recognise handwritten character images, a multi-layered feed forward artificial neural network was used as a classifier. Some preprocessing methods are used to preprocess the character images before labelling, such as thinning, foreground and background noise reduction, cropping, and scale normalisation. But the accuracy achieved by this work is very less compared to other works.

Handwritten recognition is the subject of much study in a variety of areas. Support Vector Machine (SVM), Random Forest, and K-nearest neighbours (KNN) (Shamim *et al* 2018) are some of the classification models that have generated excellent results on small datasets and acceptable results on large datasets. The device is designed to function best with typed text rather than handwritten text. It is costly to build such structures. Also there is still some space for improvement in the existing schemes.

3. Proposed System

This system is implemented in MATLAB R2014B. Our proposed model uses ANN (Artificial Neural Network) for recognition of handwritten text in the form of images. Set of images are passed to the system as training input. Preprocessing is done on the training image dataset to remove noise. Feature extraction from the input images (Hallale & Salunke 2013) is done using Histogram of Oriented Gradients. The training is done using various handwritten numbers/characters and our model is created. Then the test image has been passed and the number/character has been predicted by comparing the trained model with test features. Finally, the accuracy is calculated by using TP (True positive), FP (False Positive), TN (True Negative) and FN (False Negative) values of the confusion matrix.

The method of transforming a non-digital representation into a digital representation is known as digitization. Scanning papers or photographs and storing them in a digital format, for example. Handwritten text is recognised and digitised in our proposed scheme using a Histogram of Gradients and an Artificial Neural Network.

3.1. Dataset description

First, the MNIST dataset was used to recognise handwritten digits and numbers. 60,000 examples in the training set and 10,000 examples in the test set are available in the MNIST database of handwritten digits (SubbaRao Gogulamudi *et al* 2020). It's a subset of the NIST's greater collection. In a fixed-size graphic, the digits have been centered and size normalized. Binary images of handwritten digits from NIST's Special Database 3 and Special Database 1 were used to construct the MNIST database. (Ahlawat *et al* 2020). SD-3 was initially designated as the instruction range, while SD-1 was designated as the evaluation set by NIST. In contrast to SD-1, SD-3 is much cleaner and easier to recognise. The explanation for this is that SD-3 was gathered from Census Bureau workers, while SD-1 was obtained from high school students. Drawing reasonable conclusions from learning tests necessitates that the outcome be independent of the instruction and test sets chosen from the whole collection of samples. As a result, a new database was created by combining NIST datasets.

As next, a dataset of some handwritten numbers and characters from different people has been created and collected as images. By using this dataset, our proposed model is trained. Some handwritten sentences created by different people are also collected for training and testing purposes.

Table 1. Dataset details

Dataset	*No. of Training Samples*	*No. of Testing Samples*	*Total no. of Samples*	*No. of classes*
Digit	60100	10020	70120	10
Alphabet	260	40	300	26

Table 1 provides the details of the dataset used for our system. In this table, the details of number of training samples, number of test samples, total number of samples and number of classes is included. The digit dataset includes both the MNIST digits as well as the handwritten real world digits. The alphabet dataset has different handwritten samples of alphabets collected from different people. Our system is trained with 60100 digit samples and 260 alphabet samples. Our dataset has the testing samples of around 10020 digits and 40 alphabets. The number of classes in our dataset is totally 36, for digits 10 classes (from 0 to 9) and for alphabets 26 classes (from A to Z).

3.2. Architecture of proposed system

In our work, handwritten numeral and character recognition using deep learning is proposed that uses HOG (Histogram of Gradients) for feature extraction and Artificial Neural Network for prediction of results. The process flow of our proposed system is represented in Figure.2. It has totally 6 phases such as data processing, threshold selection, detection, cropping and resizing, feature extraction, prediction and accuracy calculation.

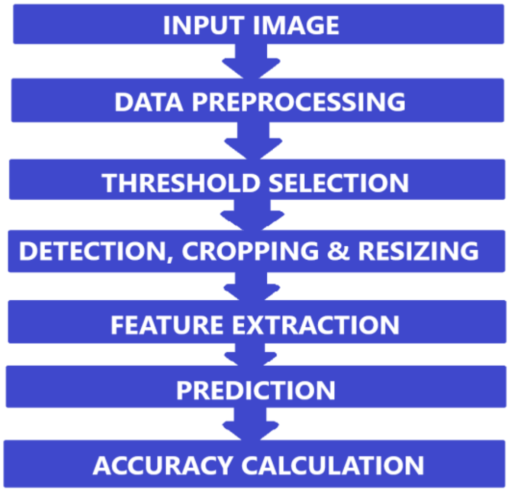


Figure 2. Process flow of the proposed system

3.3. Modules

3.3.1. Data preprocessing

Data Preprocessing is the method of transforming or encoding data so that it can be easily parsed by the computer. It needs to be done to make the process easier. Preprocessing is done using morphological operations. Morphological operations add a structuring element to an input image and produce a similar-sized output image. In a morphological method, the value of each pixel in the output image is calculated by comparing it to its neighbours in the input image.

3.3.2. Threshold selection

The picture has been converted to grayscale (as seen in real life) and a threshold value has been calculated. The threshold is set to a range of [0, 1]. Otsu's approach selects a threshold that minimises the intraclass variation of the black and white pixels that have been threshold. By replacing all pixels in the input image with luminance greater than level with the value 1 (white) and all other pixels with the value 0 (black), the grayscale image has been transformed to a binary image (black) (Choudhary *et al* 2013).

3.3.3. Detection, cropping and resizing

The region of text area is detected and cropping of each number and character has been done by Bounding box which is in regprops method. Each number is resized by 100x50.

3.3.4. Feature extraction

The features from the input images i.e., numbers and characters are extracted using HOG. The feature descriptor HOG (Histogram of Oriented Gradients) is frequently used to extract features from image data. It is commonly used in object detection tasks in computer vision. The HOG descriptor is concerned with an object's structure or form.

3.3.5. Prediction

The test image is transferred to the machine to test the outcome after the training phase is completed. An artificial neural network performs the prediction (ANN). The Artificial Neural Network (ANN) uses the brain's computation to create algorithms that can be used to model complex patterns and solve prediction problems.

3.3.6. Accuracy calculation

The percentage of correct predictions made by our model is called accuracy. Accuracy has been calculated by TP (True positive), TN (True Negative), FP (False Positive), FN (False Negative). Accuracy has been calculated using the Equation (3.1)

$$Accuracy \ = \frac{TP \ + TN}{TP + TN + FP + FN} \tag{3.1}$$

Sensitivity is the ability to correctly identify the true positive rate. Sensitivity has been calculated using Equation (3.2) as the number of correct positive predictions (True positive) divided by the total number of positives.

$$Sensitivity = \frac{TP}{TP + FN} \tag{3.2}$$

Specificity is the ability to correctly identify the true negative rate. Specificity is calculated using Equation (3.3) as the number of correct negative predictions (True Negative) divided by the total number of negatives.

$$Specificity \ = \frac{TN}{TP + FN} \tag{3.3}$$

4. Results and Discussion

The sample handwritten images collected from different people are given as input. These images are preprocessed and trained and the results are predicted.

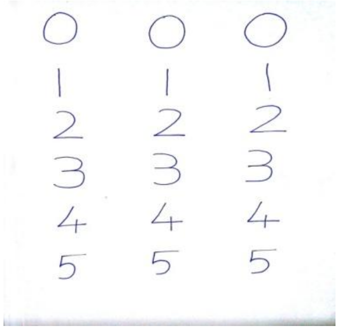


Figure 3. Input image of handwritten digits

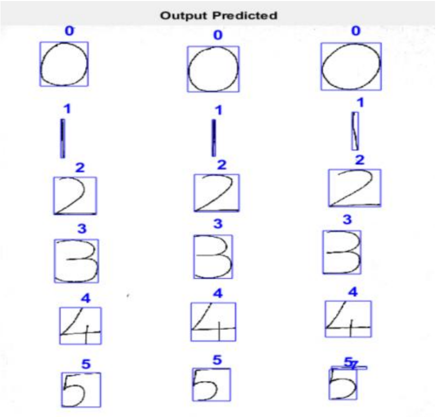


Figure 4. Output image of handwritten digits

After training with our dataset has been done, an image of handwritten numbers from 0 to 5 (Figure 3) is given as input to our system for testing. The result predicted by our system is shown in Figure 4.

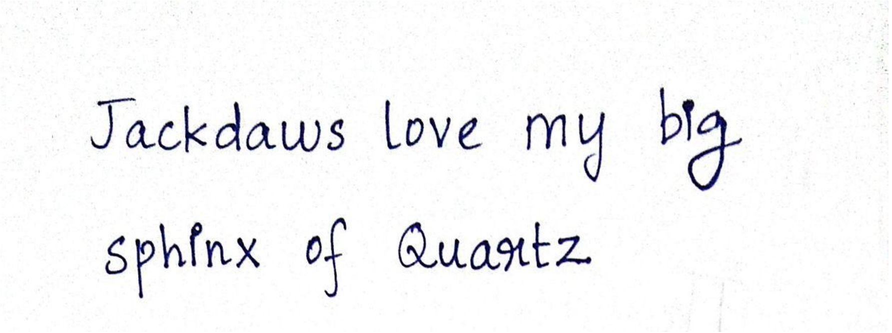


Figure 5. Input image of handwritten sentence

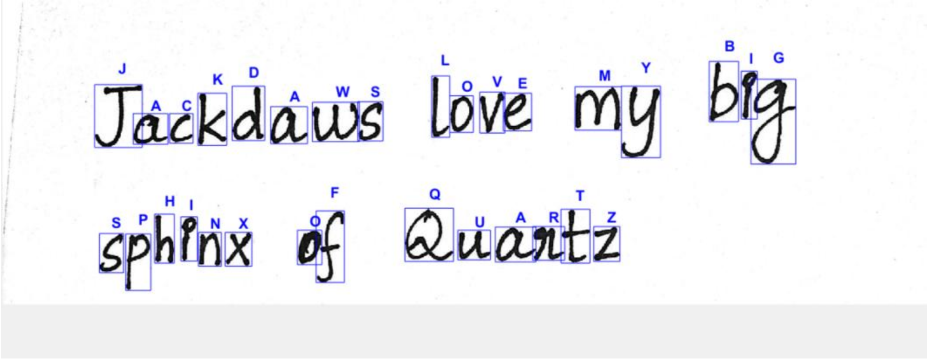


Figure 6. Output image of handwritten sentence

An image of the handwritten sentence “Jackdaws love my big sphinx of Quartz” (Figure 5) is given as input to our system for testing. The result predicted by our system is shown in Figure 6.

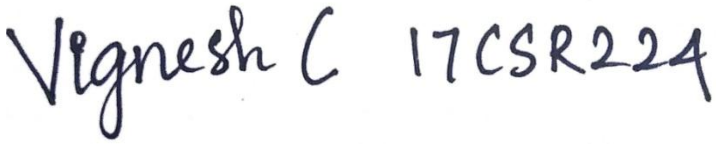


Figure 7. Input image of handwritten text with cursive letters and numbers

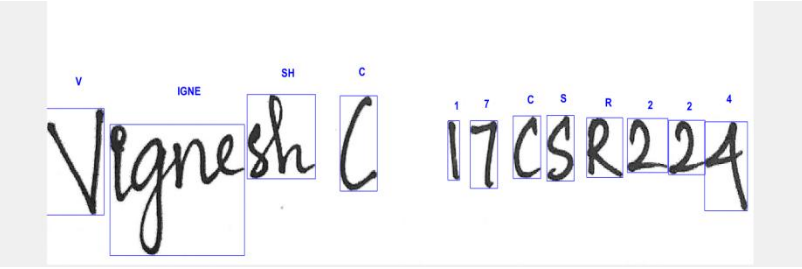


Figure 8. Output image of handwritten text with cursive letters and numbers

An image of the handwritten sentence “Vignesh C 17CSR224” (Figure 7) is given as input to our system for testing. The result predicted by our system is shown in Figure 8.

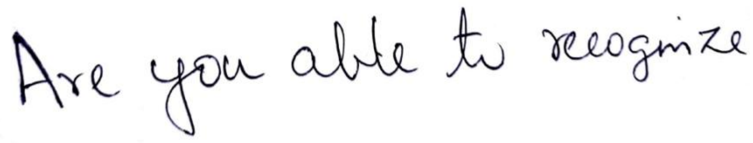


Figure 9. Input image of handwritten sentence with cursive letters

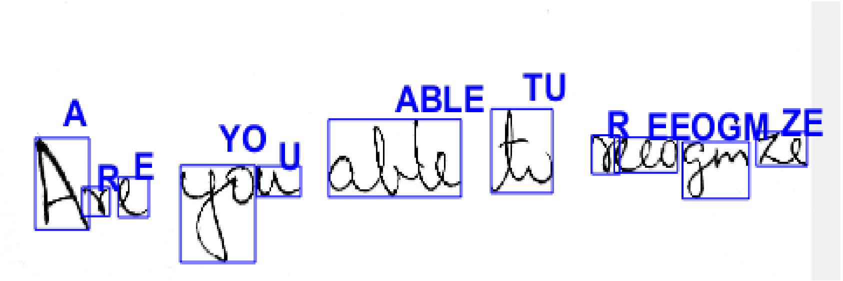
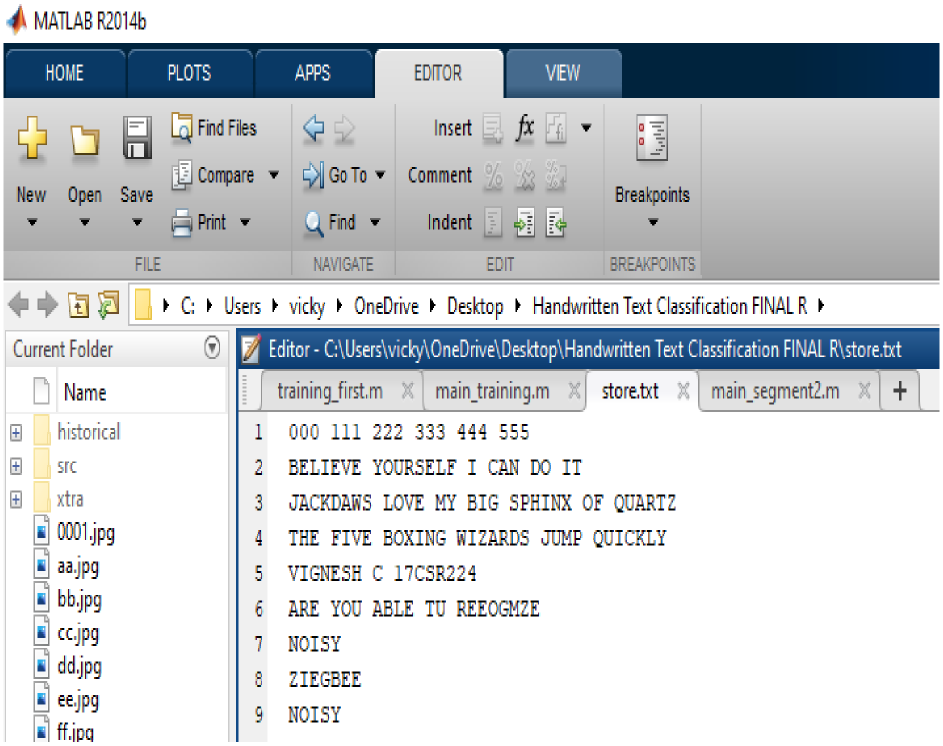


Figure 10. Output image of handwritten sentence with cursive letters

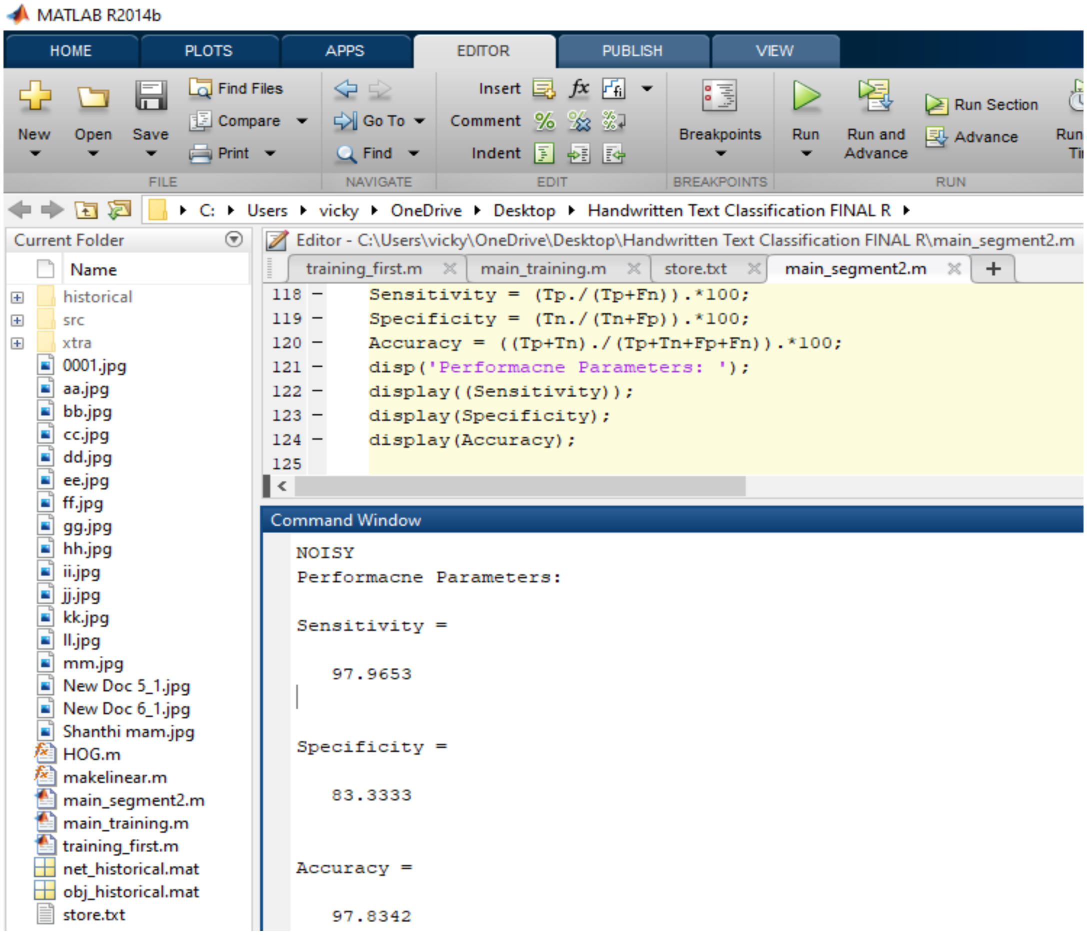
An image of the handwritten sentence “Are you able to recognize” (Figure 9) is given as input to our system for testing. The result predicted by our system is shown in Figure 10. The word ‘to’ is written like ‘tu’ and hence it is recognised as ‘tu’ only. Also for the word ‘recognize’, it is recognized as ‘reeogmze’ just because the running letters are similar to that.



```
000 111 222 333 444 555
BELIEVE YOURSELF I CAN DO IT
JACKDAWS LOVE MY BIG SPHINX OF QUARTZ
THE FIVE BOXING WIZARDS JUMP QUICKLY
VIGNESH C 17CSR224
ARE YOU ABLE TU REEOGMZE
NOISY
ZIEGBEE
NOISY
```

Figure 11. Digitization result

After testing of each and every handwritten text image, the recognized text is stored in a file store.txt. The digitization result of our system for the above mentioned images is shown in Figure 11.



```
Sensitivity = (Tp./(Tp+Fn)).*100;
Specificity = (Tn./(Tn+Fp)).*100;
Accuracy = ((Tp+Tn)./(Tp+Tn+Fp+Fn)).*100;
disp('Performacne Parameters: ');
display((Sensitivity));
display(Specificity);
display(Accuracy);
```

```
NOISY
Performacne Parameters:

Sensitivity =

   97.9653

Specificity =

   83.3333

Accuracy =

   97.8342
```

Figure 12. Performance parameters

The Performance Parameter of our system is shown in Figure.12. In this figure, performance parameters such as sensitivity, specificity and accuracy are calculated and printed. The percentage of correct predictions made by our system is 97.83%. The sensitivity of our system is 97.96%, which means there are few false negative results. The specificity of our system is 83.33% that means our system gives some false positive results also.

5. Conclusion and Future work

Handwriting recognition is very much useful in real-world applications such as the translation of handwritten information into digital format, like number plate recognition, bank check processing, postal code recognition, signature verification, etc. Our proposed handwriting recognizing system has the ability to recognise both handwritten digits and handwritten characters including most of the cursive letters. As this system automatically adds the recognized content into a text file as a form of digitization, this system is unique compared to the other existing handwritten recognition systems. This feature is mainly useful in educational institutions for handwritten question paper recognition and digitization. This system is trained with only a few people’s handwriting and if this is extended with many people’s handwriting this system can perform remarkably in such question paper recognition and digitization. The accuracy achieved by our model is 97.83%.

In future, we plan to create a model that can provide higher accuracy even for images with more noise. Accuracy can be furthermore improved by collecting a large number of handwritten image samples from different people. Thus, by further optimizing the model, we are expecting to minimize the loss percent.

References

1. Ahlawat, S., Choudhary, A., Nayyar, A., Singh, S., & Yoon, B. (2020). Improved handwritten digit recognition using convolutional neural networks (CNN). *Sensors*, 20(12), 3344.
2. https://www.mdpi.com/1424-8220/20/12/3344
3. Ali, S., Shaukat, Z., Azeem, M., Sakhawat, Z., Mahmood, T., & ur Rehman, K. (2019). An efficient and improved scheme for handwritten digit recognition based on convolutional neural networks. *SN Applied Sciences*, 1(9), 1125.
4. https://link.springer.com/article/10.1007/s42452-019-1161-5
5. Bora, M. B., Daimary, D., Amitab, K., & Kandar, D. (2020). Handwritten Character Recognition from Images using CNN-ECOC. *Procedia Computer Science*, 167, 2403-2409.
6. https://www.sciencedirect.com/science/article/pii/S1877050920307596
7. Choudhary, A., Rishi, R., & Ahlawat, S. (2013). Off-line handwritten character recognition using features extracted from binarization technique. *Aasri Procedia*, 4, 306-312.
8. https://www.sciencedirect.com/science/article/pii/S2212671613000462
9. Hallale, S. B., & Salunke, G. D. (2013). Twelve directional feature extraction for handwritten English character recognition. *International Journal of Recent Technology and Engineering*, 2(2), 39-42.
10. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.677.2586&rep=rep1&type=pdf
11. Manchala, S. Y., Kinthali, J., Kotha, K., Kumar, J. J. K. S., & Jayalaxmi, J. (2020). Handwritten text recognition using deep learning with Tensorflow. *International Journal of Engineering and Technical Research*, 9(5).
12. https://cutt.ly/ibwz1sm
13. Qiao, J., Wang, G., Li, W., & Chen, M. (2018). An adaptive deep Q-learning strategy for handwritten digit recognition. *Neural Networks*, 107, 61-71.
14. https://www.sciencedirect.com/science/article/pii/S0893608018300492?casa_token=mtoUCCxn634AAAAA:ru4MDbLhuf9QiQZ7ZOLUFV1bv0EClTndDlwhwBq2gdri_ASmpE9XTbnQfSIUS6PpGuVhl6Fm9Q
15. Shamim, S. M., Miah, M. B. A., Angona Sarker, M. R., & Al Jobair, A. (2018). Handwritten digit recognition using machine learning algorithms. *Global Journal Of Computer Science And Technology*.
16. SubbaRao Gogulamudi, Vital Kumar Pinnela, Lakshmi Sai Tejaswi Pathuri, RamTeja Borra. (2020). Handwritten Digit Recognition by using Pattern Recognition & Consensus Clustering. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* ISSN: 2278-3075, Volume-9 Issue-6.
17. Trivedi, A., Srivastava, S., Mishra, A., Shukla, A., & Tiwari, R. (2018). Hybrid evolutionary approach for Devanagari handwritten numeral recognition using Convolutional Neural Network. *Procedia Computer Science*, 125, 525-532.
18. https://www.sciencedirect.com/science/article/pii/S1877050917328326