International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Novel Approach for Hand-Written Digit Classification Using Deep Learning

Tamanna Sachdeva[1], Nisha Jha[2,] Geeta [3], Hemalatha Thanganadar[4], Mostafa M. Salah[5], Shaurya Deep[6], Chaman Kumar[7]

Submitted: 16/09/2023 **Revised**: 19/11/2023 **Accepted**: 29/11/2023

Abstract: Humans' control over technology is at an all-time high, with applications ranging from visual object recognition to the dubbing of dialogue into silent films. Using algorithms for deep learning and machine learning. Similarly, the most crucial technologies are text line recognition fields of study and development, with an increasing number of potential outcomes. Handwriting recognition (HWR), also identified as Handwriting Text Acknowledgment, is the capacity of a computer to understand legibly handwritten input from bases such as paper documents, screens, and other devices. Evidently, we have performed handwritten digit recognition using MNIST datasets and SVM, Multi-Layer Perceptron (MLP), and CNN models in this research. Our primary purpose is to compare the accuracy and execution times of the aforementioned models to determine the optimal model for digit recognition.

Keywords: *Handwriting recognition, Support Vector Machine, Multi- Layer Perceptions, CNN.*

1. Introduction

Artificially intelligent picture analysis is a fascinating field of study. Handwritten digits acknowledgment is a well-researched subfield in science that focuses on the development of the process of teaching a model to tell apart written digits that have already been divided. The importance of this issue cannot be overstated in the fields of data mining, machine learning, and pattern recognition, as well as several other fields of artificial intelligence. Over the past decade, machine learning techniques have been put to use primarily, resulting in systems that are competitive with human performance and vastly superior to the manually developed classical AI systems utilized in the early days of visual character acknowledgment technology. Though, not all characteristics of these particular models have been evaluated in the past. Researchers in machine learning have made significant efforts to devise efficient algorithms for approximating data-based recognition [3]. Handwritten digital communication has its own standard in the twenty-first century, and it is frequently cast off as a means of interaction and capturing info to be collected with others. Since various groups may use distinct kinds of writing and draw the same design for the letters of their recognized script, it might be challenging to identify handwritten text due to the diversity and distortion of the handwritten writing system. One of the most pressing problems in the expansion of digit recognition systems is the selection of the digit from which the greatest cultivated features may be obtained. In pattern recognition, many types of region sampling approaches are employed to identify such regions [4]. The difficulty in handwritten appeal documentation is mostly due to the wide variety of individual writing styles [5]. Consequently, effective feature extraction is crucial for enhancing the presentation of a handwritten atmosphere acknowledgment scheme. Due to its widespread use, handwritten digit identification has attracted significant attention in the field of pattern recognition systems in recent years. Existing paper documents might be digitized and processed using character recognition technology in the near future, therefore laying the groundwork for a paperless environment. The nature of handwritten digit datasets is ambiguous since the lines are not always crisp and absolutely straight. One of the primary objectives of feature extraction in digit recognition is to eliminate redundant information and provide a more accurate representation of the word picture through a collection of numerical properties. It focuses on obtaining the majority of critical information from picture raw information [6]. Additionally, the arcs are not unavoidably as flat as those

[1] *Assistant Professor Department of Computer Science and Technology ,Manav Rachna University Faridabad, 121003 Haryana, India E-mail tamannasachdeva26@gmail.com*

[2] *Assistant Professor, Department of Information Technology, Jagannath International Management School, Vasant Kunj, New Delhi-110070, India.*
Email: nishajha1992@gmail.com

[3] *Scholar of CSE, Northcap University, Sector 23 Gurugram,India.*
Email: geeta22csd001@ncuindia.edu

[4] *Department of Health Informatics, College of Public Health and Tropical Medicine, Jazan University, Jazan, 45142, Kingdom of Saudi Arabia. Email: hthanganadar@jazanu.edu.sa*

[5] *Assistant Professor Electrical Engineering Department, Future University in Egypt, Cairo 11835, Egypt*
Email- mostafa.abdulkhalek@fue.edu.eg

[6] *ASSISTANT PROFESSOR DEPARTMENT:- BCA IIMT COLLEGE OF MANAGEMENT, GREATER NOIDA**
EMAIL:- shauryadeep68@gmail.com

[6] *Assistant professor iimt college of engineering Greater Noida*
Email- chamankumar31@gmail.com

of written letters. In addition, characters in a dataset might be drawn in varying sizes and orientations, even if they are always intended to be vertically or horizontally printed on a guideline. Consequently, an effective handwriting recognition system may be constructed by taking these restrictions into account. Occasionally, it may be rather laborious to identify hand-written characters, given that the majority of humans cannot even recognize their own scripts. That's why it's important for a writer to make their handwriting look authentic if they want their work to be accepted as such. The software engineering component is presented first, before detailing the research approach that was really used. The article [7] presents several categorization strategies for feature extraction, such as structural feature-based algorithms.

2. Literature Survey

The author [1] tried to set up a primary purpose to locate a representation of individual handwritten digits that enables efficient recognition of such digits. The authors gave a research experiment with several different machine learning algorithms because it's necessary for reading handwritten numbers. In any kind of recognition procedure, the most critical challenge is figuring out how to get the proper methods for extracting features and classifying them. When developing the suggested strategy, we tried to account for both time and factors to takes to complete the task. During the recognition process, the Multilayer Perceptron was able to obtain the highest level of accuracy possible, which was 90.27%. The purpose of this effort, which was being carried out as a preliminary attempt, purposed to aid in the reading of handwritten digits without resorting to pre-existing categorization schemes.

The author of [2] has developed three models Specifically, it is used for recognizing handwritten numbers, using MNIST datasets. These models were based on the use of various techniques. In order to determine which of these models was the most accurate, they compared them based on the traits they shared. However, because of its relative simplicity, support vector machines were unable to classify ambiguous and complex images with the same degree of precision as was possible with MLP and CNN algorithms. Since they relied on one of the most fundamental classifiers, support vector machines could complete their tasks much more quickly than most other algorithms. In this particular scenario, they give the highest training accuracy rate. According to our research, CNN produced the best accurate results when attempting to recognize handwritten digits. Because of this, the author was forced to the conclusion that CNN was the most effective solution for any kind of prediction issue that utilizes pictures the entry of data. Then, it determined that expanding the number of epochs by altering the settings of the algorithm was meaningless owing to the constraints of a certain model by comparing the times at which each method completed the work. The author also discovered that after a certain amount of repetitions, the model started overfitting the dataset and making inaccurate predictions.

Initially, the author [3] utilized extra pre-processing techniques, like as jittering, to provide training that was more engaging and robust. To normalize the data, the author divided each pixel by its matching standard deviation. In order to rapidly analyse and improve our model, we limited the test to 20 training instances for each given word due to time and financial constraints. A second technique for enhancing our character segmentation model would be to abandon the pursuit of the most probable solution. The author would address this problem by contemplating an exhaustive yet yet effective decoding algorithm, such as beam search. The test might utilise a character/word-based language-based model to assign a penalty/benefit score to each of the final beam search candidate routes, in addition to their combined individual softmax probabilities, which represent the probability of the sequence of characters/words. If the language model shows that the most probable candidate term according to the softmax layer and beam search was highly improbable given the context thus far, then the model could self-correct.

The author [4] compared several algorithms for handwritten number recognition using the MNIST database. Numerous scientists have developed innovative ways for handwritten digit identification that have made our lives simpler throughout time. In this research, the accuracy and computing time both rise as the number of system layers grows. Despite much study and development, systems proved unable of competing with human intellect. After examining numerous classifiers, the authors conclude that the classifier ensemble presented by Rafele et al[10] has the highest accuracy but requires a significant amount of processing time, and that the best approach for handwritten number identification is 6-layer NN with the lowest error rate.

The author [5] discussed the many issues associated with text subdivision by giving a set of methods that might be enhanced in upcoming research. The divided pieces might enter the recognition step in order to convert handwritten characters to printed ones because Arabic is written with ligatures and a cursive style,, it has proven impossible to utilise a generic formula to construct text parts. After several experiments, 14 CNN designs were recommended for the first experimental route. Training was conducted using the HMBD dataset. As a measure of performance, testing precision was utilised to determine the optimal architecture. Various optimization techniques were also utilised to improve CNN performance.

The author [6] provided a comparison of the digit recognition capabilities of many deep learning algorithms, including multilayer (CNN) using Keras and various technologies. Keras is a Python library for high-level neural networks that is intuitive enough for beginners to pick up and use. Theano, is consist open-source software libraries that enable high-performance numerical computing. Its design was so adaptable that it could be simply implemented on a wide variety of platforms, including (CPUs, GPUs, and TPUs), as well as desktop computers, clusters of servers, and smart handheld devices. After conducting more research and making comparisons about the accuracy achieved by the aforementioned algorithms, the findings appear to be as follows: Convolutional neural network accuracy was 90.69 percent, SVM 90.89 percent, KNN 91.67 percent, and (KNN) 86.89 percent for number recognition (RFC). As a result, it is now abundantly obvious that when compared to alternative approaches, convolutional neural networks yield superior outcomes in terms of both prediction accuracy and overall performance.

Using various deep learning models, the author [8] created an organization of handwritten city names and HTR. The following identification accuracy data were obtained from tests using a range of machine learning approaches applied to handwritten city names in the categorization task. When applied to the final rectification of the text under recognition, Wordbeamsearch's usage of a dictionary yielded the best results. The following recognition accuracy values were also obtained from experiments on HTR using various deep learning approaches.

In this paper, the author [9] proposes a new dataset of difficult Arabic digits collected from schools with varying degrees of education. Distributing and collecting digital forms from hundreds of elementary, secondary, and college students resulted in the collection of a big dataset. After the author discovered that there were few and unchallenging Arabic digit datasets, we exerted great effort to compile such a dataset. In addition, the gathered dataset was trained with a CNN model that reflects the state of the art for a number of applications. Thus, we exhaustively examined the model by picking their parameters with care and demonstrating its resilience in dealing with our dataset.

The author [10] described an approach that creates a deep neural network by stacking the concealed coats of k-sparse encoders. After training the system on a more comprehensive, we tested its ability to recognize individual handwritten digits and digit strings from the CVL data sets. In the context of a broader system, the proposed approach was developed to analyse the key information like amounts and messages. To prevent overfitting, the first two layers of the network were trained independently to hierarchically relevant features rather than applying the back propagation approach directly on randomised weights. The training set was expanded by include the pictures that were rebuilt with varying settings. Future work will involve expanding the system to work on numeric and text strings directly on check pictures via synthetic string creation. Using changed training procedures and modules to enhance the effectiveness of feature representation is another area of future research.

The author[11] compares several approaches for handwritten number recognition using the MNIST database. Numerous scientists have developed innovative ways for handwritten digit identification that have made our lives simpler throughout time. In this research, the accuracy and computing time both rise as the number of system layers grows. Despite much study and development, systems proved unable of competing with human intellect. After analysing numerous classifiers, the authors conclude that the suggested classifier ensemble is the most accurate but also the most computationally intensive, and that the optimal approach for handwritten number identification is 5-layer NN with a 0.39 percent error rate.

To circumvent the need for elaborate procedure inherent in conventional recognition systems, the author [12] analyzed variants of a CNN. The present study proposes the involvement of numerous hyper-parameters based on a comprehensive examination of an MNIST dataset. The author also confirmed that hyper-parameter fine adjustment was crucial for enhancing CNN architecture performance. With the Adam optimizer, we got a 93.89% recognition rate, which is superior to any previously published values. The originality of this study is that it explores exhaustively all the constraints of CNN building that provide the accurate result. Competitors were unable to match this accuracy using a purely CNN model. The present study's accuracy was comparable to that of previous research that used ensemble CNN network topologies for the same dataset to improve recognition accuracy, but at a higher computational cost and with more sophisticated testing.

The author [13] has experimented with several machine learning techniques and data training models detection and achieves the maximum accuracy in predicting handwritten number. Thus, the author decided to categorise a given image of a handwritten number as the needed digit using five distinct methods and then assess its accuracy. The author developed handwriting recognizers, analysed their performance on the MNIST dataset, and subsequently enhanced training speed and recognition performance. In addition to developing a system for word-based handwriting recognition, it is necessary to test the handwriting of a given word and identify the writer by picking which training sample was recognised by the majority of users. The author elaborated on the recent developments in handwritten character recognition. The most precise answer in this area is directly or indirectly dependent on the kind and quality of the information to be read. This article describes many strategies for character recognition in handwriting recognition systems.

The classification was the output of the developed deep neural network (DNN). In order to identify unfamiliar handwritten numbers, the author [14] proposed a method based on ghost imagery generated by the cosine transform speckle filter. As the sample ratio increased, the accuracy of recognition improved. The proposed strategy gives marginally improved performance with less complexity and non-locality in comparison to the standard recognition method applying the same DNN structure. The proposed technique is an effective remote sensing method. The classification was the output of the developed deep neural network (DNN). The author [15] provided identification technique for indefinite handwritten digits based on ghost imagery created by the cosine transform speckle. Simulations indicate that A higher degree of accuracy in recognition is achieved using the suggested strategy by up to 81%. As the sample ratio increased, the accuracy of recognition improved. The proposed strategy gives marginally improved performance with less complexity and non-locality in comparison to the standard recognition method applying the same DNN structure. The proposed technique is an effective remote sensing method.

3. Materials and Methods

2.1 Multilayer Insights

To categorise the handwritten digits, a neural network-based classifier known as (MLP) is utilised. The contribution layer, the unseen layer, and the yield layer are the three parts that make up a multilayer perceptron. Each layer can have any amount of nodes, and each neuron in a given layer connected, to those in all higher layers[12]. Consequently, the feed forward network is another name for it. There will be as many nodes in the input layer as there are attributes in the dataset. The number of nodes in the output layer is proportional to the number of obvious classes in the dataset. It is difficult to establish the optimal number of hidden layers or the best number of nodes in a unseen layer for a given task. In general, though, these quantities are chosen empirically. The link between two nodes in a multilayer perceptron consists of a weight. During the training phase, the robot essentially studies the precise weight adjustment that resembles to each link [13]. Back propagation method is used as a supervised learning approach for learning purposes.

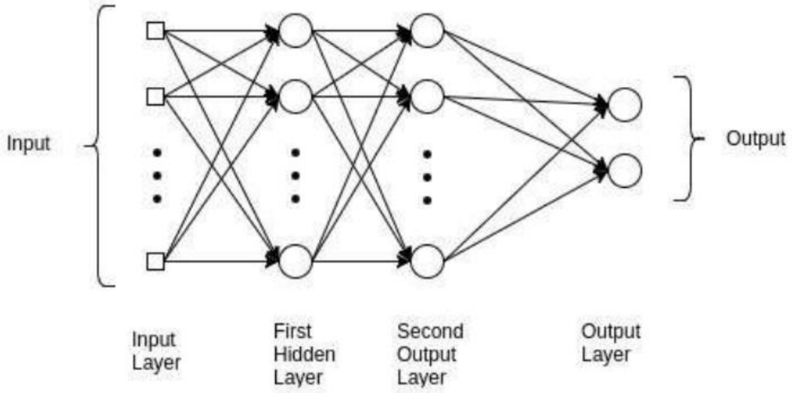

Fig 1: The Multilayer Perceptron with Variable Network Specification is depicted in this figure's fundamental architecture.

2.2 Support Vector Mechanism

SVM, is a sort of oversaw techniques designed to categorize data points by maximising the boundary between modules in a high-dimensional space [15]. SVM uses a spatial representation of data in which instances are points and classes are separated by as large a distance as possible. An optimal algorithm is created through two stages: a "training" stage where an operator uses training data to create an algorithm that can discriminate between groups previously described by the operator (for example, patients and controls), and a "testing" stage where the technique is used to blind-predict the group to which a gain knowledge belongs. In addition, it gives a very precise cataloguing presentation over the training records and generates sufficient search space for the categorization of future data parameters. Therefore, it always guarantees a sequence of parameter mixtures on no less than a meaningful fraction of the data. Large datasets should be approached with caution, since they may result in an upsurge in exercise time.

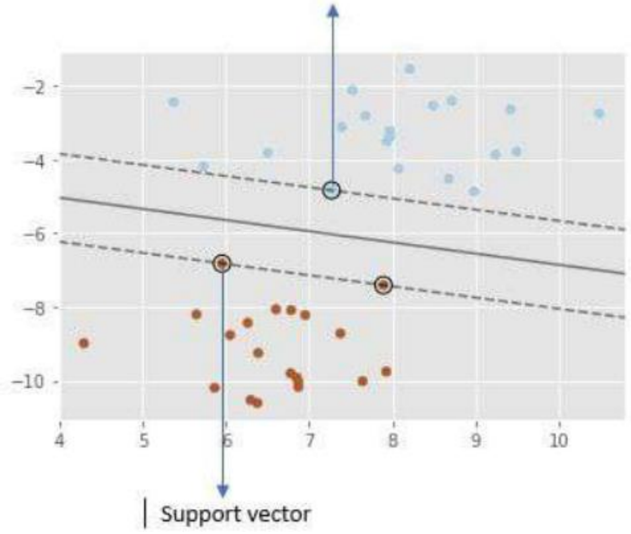

Fig 2: This picture demonstrates the supporting vectors and hyperplanes used in SVM classification.

2.3 J48

To Together with WEKA, the J48 procedure is created for the MONK1 project. The algorithm is a modification of the decision tree. There are several alternatives for tree clipping in the case of the J48 algorithm. WEKA's handy categorization algorithms attempt to clarify or prune findings. This strategy will enable us to provide more general findings and may also be used to address any overfitting concerns. Therefore, this will contribute to the correctness, despite the fact that numerous regulations will be established. However, a model's accuracy on training data will suffer if it undergoes pruning. This is due to the fact that pruning employs a number of techniques to reduce the decision tree's sensitivity. in order to boost its performance on test data. Generalizing the algorithm until it achieves a balance between precision and adaptability is the entire premise. The J48 utilises two pruning techniques. This procedure starts at the mature tree's leaves and aims to reach the tree's roots. Subtree rising is the second type of pruning used. There is typically no obvious method to predict the value of the option, however it may be advantageous to disable it if the induction process is taking an excessive amount of time. Error rates are necessary for drawing accurate judgments on which tree branches to raise or replace. There are several ways to do this task. The most straightforward method is to set aside a part of the training data for decision tree testing. To prevent overfitting, the decision tree can be tested with the information set aside for this purpose. "reduced-error pruning" describes this method. For particularly tiny datasets, avoiding decreased error trimming may be advantageous.

2.4 Random Trees

An unpruned regression or learning algorithm ensemble called a "random forest" is activated with iterations of the exercise data and a chance feature assortment method. The forecast is made by aggregating the ensemble's classification predictions via superiority voting. It is more sensitive to background noise and so yields the generalisation error rate. The fact that it's intended to improve accuracy generally means that it will prioritise the show considerably of the majority class over that of the minority, leading to continuously low precision for the latter.

2.5 Dataset Description

The recognition of handwritten digits is a large research subject that provides a full overview of the field, covering important feature sets and methods. In conflicting to visual character recognition, which is optimised for reading printed text where unique fonts are available and where there is little variation in character size, font, and other features, handwritten character recognition is designed to read text that has been written by hand. For character recognition systems, several feature extraction strategies have been presented. Using approaches such as, the challenges encountered in handwritten numeral recognition have been analysed. Numerous languages, including English, Chinese, Japanese, and Arabic, have been the focus of expanded efforts to improve the identification of digits. In India, numerical recognition mostly focused on Devanagari, Tamil, Telugu, and Bengali. In our experiment, we utilised a dataset of digits given by the Austrian intelligence. Based on this data set, it seems that the Mitchell down-sampling filter with a blur value of 4.5 and a resolution of 10x11 pixels works well, should be down-sampled.

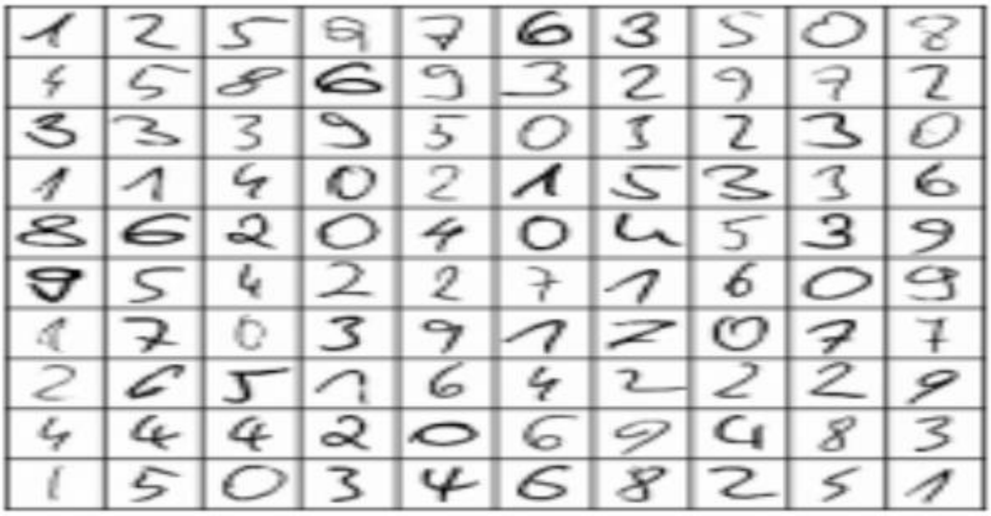

Fig 3: A part of a handwritten sample dataset

4. Implementation

In order to compare the algorithms' working accuracy, execution speed, complexity, and number of epochs (in deep learning algorithms), we employed three distinct classifiers:

- Support Vector Machine Classifier
- ANN - Multilayer Perceptron Classifier
- Convolutional Neural Network Classifier

3.1 Pre-Processing

The first phase in machine learning and deep learning is pre-processing, which aims to improve the input data by removing undesired impurities and redundancies. All of the photos in the dataset were moulded into 2-dimensional images in order to streamline and divide the input data (28,28,1).

Since the picture pixel values vary from 0 to 255, we normalised them by converting the dataset to a "float32" format and dividing by 255.0 to make the input features range from 0.0 to 1.0. The y values were then one-hot encoded into zeros and ones, categorising each number. For instance, the output value 4 will be turned into an array of zero and one values, [0,0,0,0,1,0,0,0,0,0].

3.2 Support Vector Machine

Both dense (numpy.ndarray, which may be converted to that by numpy.asarray) and sparse (any scipy.sparse) sample vectors are supported as inputs for the SVM in scikit-learn. SVC, NuSVC, and LinearSVC are classes in scikit-learn that can perform multi-class classification on a dataset. We employed LinearSVC in this study to classify MNIST datasets that use a linear kernel that was constructed with the aid of LIBLINEAR.

For the implementation, a variety of scikit-learn libraries including NumPy, matplotlib, pandas, Sklearn, and seaborn were employed. We'll first download the MNIST datasets, then use pandas to import and read those CSV files.

3.3 Multilayered Perceptron

With the aid of the Keras module, the Multilayer Perceptron [18], also known as a feedforward artificial neural network, is used to implement the recognition of handwritten digits. The MLP model of the Sequential class is created, and the corresponding hidden layers are added, each with a different activation function, taking a 28x28 pixel image as input. After building a sequential model, we added Drop out layers and a Dense layer with various parameters, as seen in the graphic below. Here, a block diagram is provided for your convenience.

These procedures may be used to train a neural network in Keras once you have the test and training sets of data.

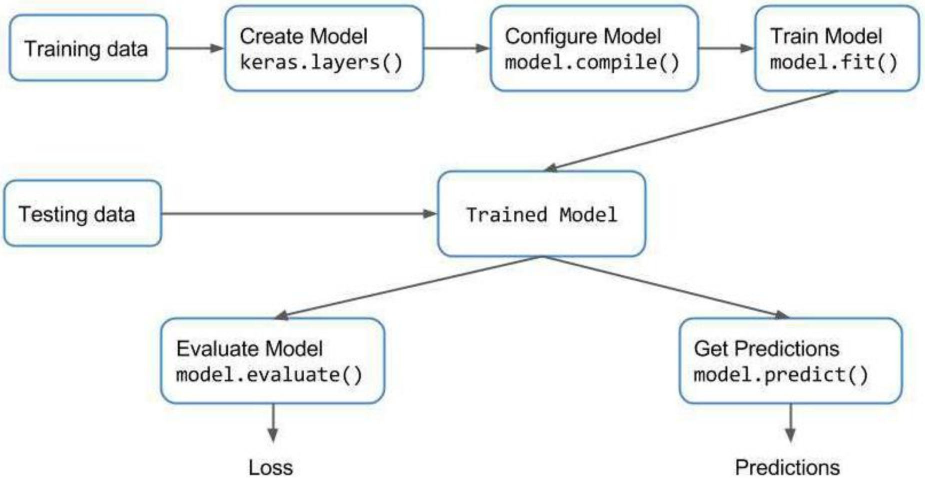

Fig 4: Multi-layers perceptron model sequential block diagram created using the Keras Module

3.3 Convolutional Neural Network

Using Keras, the Convolutional Neural Network [15] implements handwritten digit recognition. Deep learning models are created and implemented using this open-source neural network library. We utilised a Sequential class from Keras, which enabled us to build the model layer by layer. The supplied image's dimensions are set to 28(Height), 28(Width), and 1. (Number of channels). Next, we developed the model, which has a Conv layer as its initial layer [20]. In order to extract features from the input data, this layer convolves around it across its height and breadth using a matrix. The filter or kernel for this matrix is known.

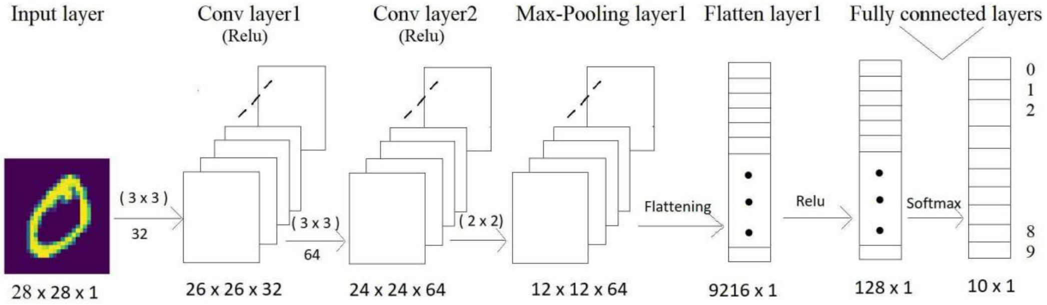

Fig 5: Convolutional neural network with a precise design and appropriate layer specs

5. Experimental Tools

WEKA is a renowned machine learning suite created by the University of Waikato and written in Java. It is open source software available. It includes a variety of procedures and visualisation tools for predictive modelling and data analysis, as well as graphical user interfaces enabling easy access to these features [30]. It covers a number of common data mining activities, including data pre-processing, classification, visualisation, clustering, feature selection, and regression. All of Weka's techniques are premised on the assumption that the data may be conveniently represented as a single flat file or relation, with each data item described by a set number of characteristics. WEKA offers many user interfaces. The Explorer is the most common entry point, however the Knowledge Flow interface, which is built on components, and the command line may provide access to the same features.

Dataset

There are numerous implementation strategies for handwritten character recognition, including popular algorithms, large learning datasets, and methods for feature extraction and features scaling. The Modified National Institute of Standards and Technology database (MNIST dataset) is a subset of the NIST dataset, which is made up of Special Database 1 and Special Database 3 from NIST. The numbers in Special Databases 1 and 3 were entered by high school students and US Census

Bureau staff members, respectively. MNIST has 70,000 handwritten digit pictures in total (60,000 for training and 10,000 for testing), each with a bounding box of 28x28 pixels with anti-aliasing. All of these photos have associated Y values that inform the viewer of the digit's identity.

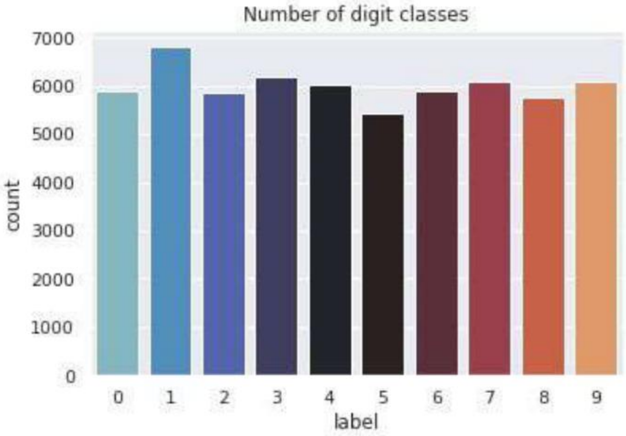

Fig 6: Bar graph illustrating the handwritten digit training dataset

6. Experimental Results and Discussion

WEKA is equipped with many graphic user crossing point that provide access to the underlying functionality. In WEKA, each dataset is referred to as an instance, and each data characteristic is referred to as an attribute. In order to facilitate analysis and assessment, the experiment's results are subdivided into different subgroups. In the first section, cases that were properly and wrongly categorised will be separated into numeric and percentage values, followed by the. For evaluation and comparison, the comparative absolute error and root relative squared error are shown as percentages in the experiment. The imitation of findings are displayed in tables 1 and tables-2 below. In table-1, we primarily describe the results of our experiment based on the precision and duration of each simulation. In addition, Table 2 displays the outcome depending on inaccuracy during the WEKA simulation.

Table 2: Result of simulation based on various mistakes

Name of Algorithm	Mean Absolute Error	Root Mean Square Error	Relative Absolute Error(%)	Root Relative Square Error (%)
Multilayer Perception	0.023	0.1231	12.78	41.04
Support vector machine	0.1511	0.273	82.38	91.05
Random forest	0.0492	0.1522	31.09	50.49
Bayes Net	0.031	0.1745	20.02	60.32
Naive Bayes	0.0351	0.0136	24.76	60.32
J48	0.0444	0.1543	23.55	65.25
Random Tree	0.0398	0.2231	25.32	70.32

The maximum accuracy based on the preceding table-1 is 92.37 percent, while the lowest is 70.06%. The average accuracy of the other algorithm is 90.81 percent. In fact, the Multilayer Perceptron classifier has the best accuracy, followed by SVM at 83.97%, RFA at 82.75%, Bayes Net at 80.35%, Nave Bayes at 85.85%, j48 at 69.51%, and Random Tree at 79.01%. Value 0 indicates absolute disagreement, whereas value 1 indicates total agreement. It verifies the accuracy of the Applying a classifier to the data collection. All important parameters to consider when comparing various classification techniques. Mean absolute error is the average of all classification algorithm errors, and the classifier with the smallest mean absolute error will be the most accurate. In table-2, Multilayer Perceptron has the lowest mean absolute error of all seven methods, at 0.023.

They found that their experiment was only sensitive to generalisations about things like cat eyes and living beings since those are abstract conceptions. They simply improve upon previous methods of classifying commonplace images criteria presents supervised and it encompasses techniques like unsupervised learning, RL, EC, and the indirect hunt for little programmes that encode deep and vast networks. They merely discussed how various pattern recognition techniques can be implemented. With a database of 10,000 examples, the highest recognition rate for handwritten Bangla characters is 76.86%, according to the results of their experiment. Online and offline recognition of handwritten Chinese characters was proposed. Their trial revealed that offline testing accuracy was highest at 79.55 percent. In our experiment, multiple machine learning algorithms were applied to recognise handwritten digits, with Multilayer Perceptron achieving the greatest accuracy of 90.37 percent.

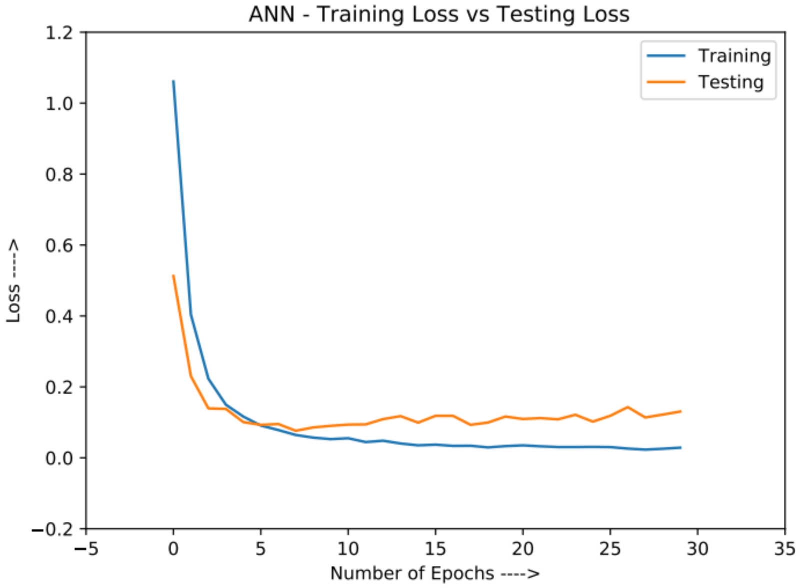

Fig 7: Loss rate v/s Number of epochs graph showing how training loss changes as the number of epochs increases in a multilayer perceptron

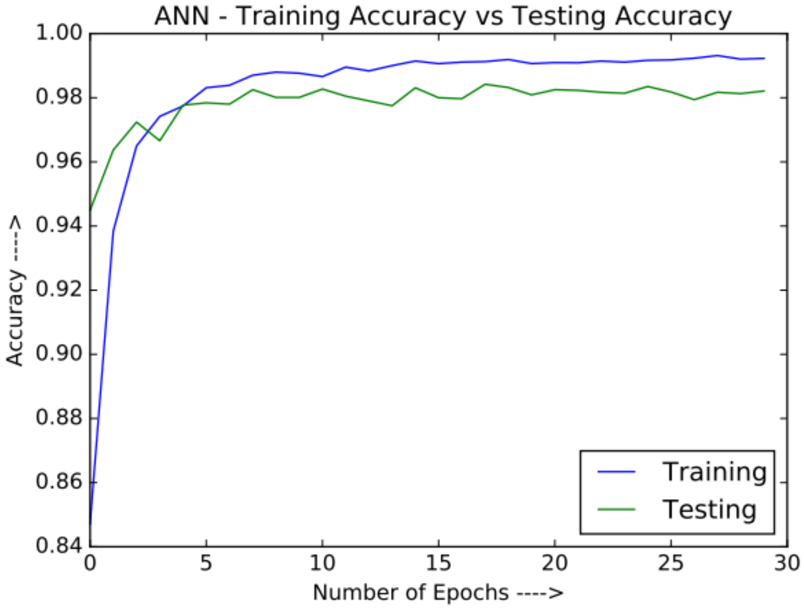

Fig 8: Accuracy v/s Number of Epochs graph showing how training accuracy changes as the number of epochs increases in a multilayer perceptron

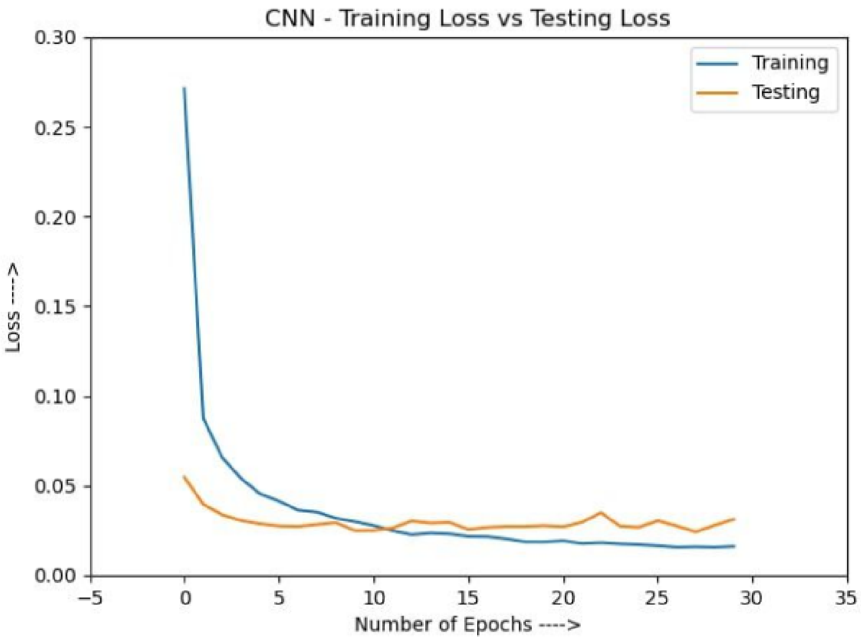

Fig 9: Loss rate v/s Number of epochs graph showing how training loss of CNN changes as the number of epochs increases

7. Future Advancements

The Applications that are built on algorithms for deep learning and machine learning have an almost endless potential for growth in the years to come. This is due to the fact that both of these categories of learning algorithms are getting more advanced as time goes on. We will be able to find answers to a wide range of issues in the not-too-distant future by designing an algorithm that is either denser or hybrid and that makes use of a greater variety of data than the collection of algorithms that is presently at our disposal. We will be able to do this by developing an algorithm that is superior in terms of functionality to the assortment of algorithms that are already on the market.

The growth of technology will, in the not-too-distant future, make it possible for us to create complex apps that are not only suitable for use by the general public, but also by the bureaucratic institutions of the highest level of government. In addition to using these algorithms for surveillance, we might, for example, put them to use in hospitals to provide more in-depth medical diagnosis, treatment, and patient monitoring. This would be in addition to the use of these algorithms for surveillance. In addition to employing them for the purpose of keeping an eye on things, there would be another use for them. The development of this field can be of assistance to us in creating an environment that is conducive to safety, attention, and comfort if we use these algorithms in both standard applications and high-level applications. This setting may be established in a shorter amount of time (i.e. applications at the corporate level or the government level). Artificial intelligence that is based on applications and deep learning are the technologies that will influence the direction that technology will go in the future. These technologies offer a degree of precision that cannot be equaled, and they bring a variety of benefits that are superior to those offered by other key challenges.

8. Conclusion

The fundamental objective of this effort is to track down a photograph of each handwritten numbers that permits quick and accurate crediting of those digits. In order to successfully recognise handwritten numbers, the authors of this study utilised a variety of machine learning algorithms and conducted experiments with each of them. In any form of recognition technique, figuring out how to acquire the procedures for effectively categorising data and extracting features is the most important and difficult problem. In order to make the process more time-effective, several predefined qualities are computed for a fresh dataset that includes each and every letter. This dataset comprises all of the alphabets. After this stage, multiple different methods of categorization are used to the increased datasets. The levels of accuracy achieved by the various models are compared and contrasted. It has been discovered that making use of an ensemble of KNN models can lead to the production of results that are the most accurate possible. When it comes to the classification models, the results are provided in the form of a confusion chart matrix. This helps to make the results easier to understand. The approach that has been provided makes an effort to take into account both the components as well as the length of time necessary to do the activity. The Multilayer Perceptron is able to achieve the highest level of accuracy that can be achieved during the process of recognition, which is 90.37% of the time. This work, which is being carried out as a first attempt, is being carried out with the objective of making it easier to recognise handwritten digits without making use of any established classification systems.

Funding Statement: Authors should describe sources of funding that have supported the work, including specific grant numbers, initials of authors who received the grant, and the URLs to sponsors' websites. If there is no funding support, please write "The author(s) received no specific funding for this study."

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] 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*.

[2] Pashine, S., Dixit, R., & Kushwah, R. (2021). Handwritten digit recognition using machine and deep learning algorithms. *arXiv preprint arXiv:2106.12614*.

[3] Balci, B., Saadati, D., & Shiferaw, D. (2017). Handwritten text recognition using deep learning. *CS231n: Convolutional Neural Networks for Visual Recognition, Stanford University, Course Project Report, Spring*, 752-759.

[4] Shrivastava, A., Jaggi, I., Gupta, S., & Gupta, D. (2019, October). Handwritten digit recognition using machine learning: a review. In *2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC)* (pp. 322-326). IEEE.

[5] Balaha, H. M., Ali, H. A., Youssef, E. K., Elsayed, A. E., Samak, R. A., Abdelhaleem, M. S., ... & Mohammed, M. M. (2021). Recognizing arabic handwritten characters using deep learning and genetic algorithms. *Multimedia Tools and Applications*, *80*(21), 32473-32509.

[6] Kumar, K. (2016). *HANDWRITTEN DIGIT CLASSIFICATION USING DEEP LEARNING* (Doctoral dissertation).

[7] Nurseitov, D., Bostanbekov, K., Kanatov, M., Alimova, A., Abdallah, A., & Abdimanap, G. (2021). Classification of handwritten names of cities and handwritten text recognition using various deep learning models. *arXiv preprint arXiv:2102.04816*.

[8] Alwzwazy, H. A., Albehadili, H. M., Alwan, Y. S., & Islam, N. E. (2016). Handwritten digit recognition using convolutional neural networks. *International Journal of Innovative Research in Computer and Communication Engineering*, *4*(2), 1101-1106.

[9] Ghosh, M. M. A., & Maghari, A. Y. (2017, October). A comparative study on handwriting digit recognition using neural networks. In *2017 international conference on promising electronic technologies (ICPET)* (pp. 77-81). IEEE.

[10] Saabni, R. (2016, September). Recognizing handwritten single digits and digit strings using deep architecture of neural networks. In *2016 Third international conference on artificial intelligence and pattern recognition (AIPR)* (pp. 1-6). IEEE.

[11] Devi, R. K., & Rani, G. E. (2019, December). A Comparative Study on Handwritten Digit Recognizer using Machine Learning Technique. In *2019 IEEE International Conference on Clean Energy and Energy Efficient Electronics Circuit for Sustainable Development (INCCES)* (pp. 1-5). IEEE.

[12] Ahlawat, S., Choudhary, A., Nayyar, A., Singh, S., & Yoon, B. (2020). Improved handwritten digit recognition using convolutional neural networks (CNN). *Sensors*, *20*(12), 3344.

[13] Chen, S., Almamlook, R., Gu, Y., & Wells, L. (2018, September). Offline handwritten digits recognition using machine learning. In *Proceedings of the international conference on industrial engineering and operations management, Washington DC, USA* (pp. 27-29).

[14] He, X., Zhao, S. M., & Wang, L. (2021). Handwritten digit recognition based on ghost imaging with deep learning. *Chinese Physics B*, *30*(5), 054201.

[15] Jana, R., Bhattacharyya, S., & Das, S. (2020). Handwritten Digit Recognition Using Convolutional Neural Networks. *Deep Learning: Research and Applications*, *7*, 51.