PURPOSE-LED
PUBLISHING™

PAPER • OPEN ACCESS

Handwritten Digits Identification Using Mnist Database Via Machine Learning Models

To cite this article: Birjit Gope *et al* 2021 *IOP Conf. Ser.: Mater. Sci. Eng.* **1022** 012108

View the article online for updates and enhancements.

Handwritten Digits Identification Using Mnist Database Via Machine Learning Models

[1]**Birjit Gope, [2,*]Sagar Pande,[3]Nikhil Karale,[4]Shivani Dharmale, [5]Pooja Umekar**

[1,2] SCSE, Lovely Professional University, Phagwara, Punjab

[3,5]Computer Science and Engineering, DRGITR, Amravati, Maharashtra.

[4]Electronics and Communication Engineering, DRGITR, Amravati, Maharashtra

[1]birjit21gope@gmail.com, [2,*]sagarpande30@gmail.com,

[3]nikhil.karale10@gmail.com, [4]shivanidharmale.10@gmail.com,

[5]poojaumekar123@gmail.com

Abstract. The identification of hand-written digits is among the most significant issue in the applications for pattern detection. In many application such as postal code, check online routing bank accounts, data form entry, etc., the applications of digits recognition include the center of the issue is the need to construct an appropriate algorithm that can identify hand-written digits and that users upload through a smartphone and scanner and other digital devices. In this paper, we took a repository of MNIST, which is a sub-set of the database of NIST results. The MNIST dataset accommodates the collection of hand-written scanned images from a broader variety of NIST repository produced by hand. The method proposed in this paper is centered on numerous machine learning methods to perform hand-written digit detection that is off-line in the python language platform. The primary objective of this paper is to render hand-written digits recognition reliable and precise. For the identification of digits using MNIST many machine learning algorithms have been used including Support Vector Machine, Multilayer Perceptron, Decision Tree, Naïve Bayes, K-Nearest Neighbor, and Random Forest.

Keywords: MNIST, Hand-written Digits, Machine Learning, Deep Learning

1. INTRODUCTION

Intelligent image processing is an enticing study area in Artificial Intelligence it is also essential for a range of existing accessible research challenges. Hand-written digit identification is a well-researched sub-area of the field that discusses the detection of pre-segmented hand-written digits with learning models. It is along with several other disciplines in artificial intelligence, one of the most critical issues of machine learning, data retrieval, deep learning, and pattern recognition [1]. The major application of machine learning approaches has been effective over the last decade in conforming to definitive systems that compete with human performance and perform substantially better than traditional artificial learning methods built manually [1]. Moreover, not all the aspects of these individual models have previously been inspected.

A significant effort has been made by researchers in data mining and machine learning to achieve successful approaches to the approximation of data recognition [2]. Hand-written digits identification correspondence has its norm in the twenty-first century and is used much of the time in everyday life as a medium of discourse and capturing the details to be communicated with others. The variety and distortion of the hand-written character collection are one of the difficulties in the overall recognition of hand-written characters since different cultures will use multiple handwriting types and control to extract the characters identical patterns from their known language.
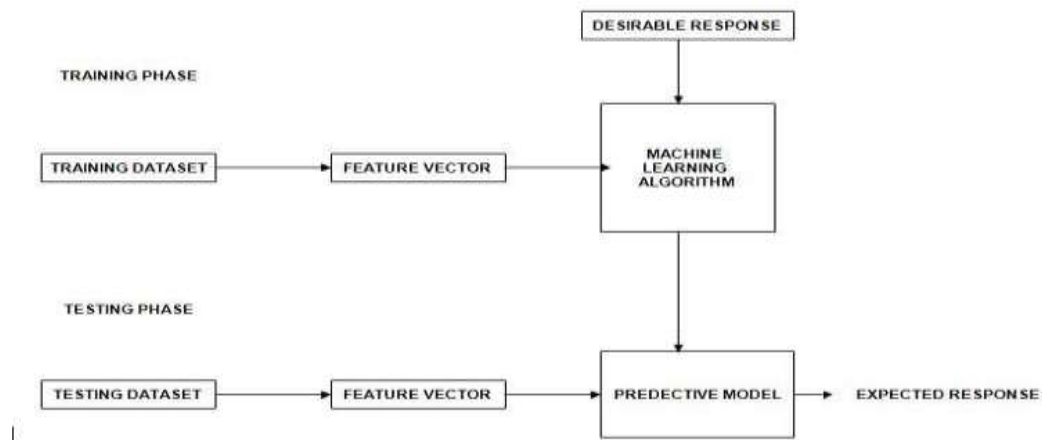
Figure 1: Phase of Identification of Handwritten digits

One of the main tasks in the area of the digital recognition system is the identification of digits from which the best discriminating characteristics can be extracted. In pattern recognition [2], various methods of area sampling strategies are used to identify certain areas. The difficulty in the identification of hand-written characters is primarily triggered by the wide variety in human writing styles [2]. To enhance the efficiency of a hand-written character recognition device, robust feature extraction is therefore quite necessary. In the field of pattern recognition device sewing to its use in different areas, hand-written digit recognition has now achieved a lot of attention. In the next few days, by digitizing and manipulating existing paper records, the character recognition technology may serve as a foundation for initiating a paperless world. Hand-written digits datasets are vague, because sharp and perfectly straight lines may not always exist. Feature extraction is the key objective of digit recognition to eliminate the uncertainty from the data and achieve a more powerful embodiment of the term symbol from a series of numerical attributes. It deals with the retrieval from raw picture details of much of the critical information [3]. In comparison, the curves, like the written characters, are not always flat. In comparison, character datasets may be drawn in multiple sizes and orientations that are often meant to be written in an upright or downright point on a checklist. Consequently, by considering these limitations, an effective hand-written recognition system can be developed. It's very exhausting to remember handwriting characters often since it can be shown that most people cannot even identify their own printed texts. Therefore, there is a restriction for a writer to compose for hand-written text appreciation.

Figure 2: MNIST datasets

Hand-written digits identification is a challenging work in a machine vision environment, and it is key to many modern technologies. The identification of hand-written digits is becoming extremely relevant in the developed world because of its realistic applications in our technological experiences. Recent years have seen the implementation of multiple recognition systems in many applications where high classification performance is needed. It lets us tackle more difficult challenges, and allows our jobs simpler. Machine learning and computer vision scientists have been commonly used to incorporate practical applications such as for the identification of zip code (postal code) an early stage hand-written digit identification has been developed. Online routing of bank accounts, the postal address is commonly used in hand-written digit identification programs [1]. A general tendency has been given to a human being to differentiate various artifacts with differences including numbers, letters, ears, speech. Executing a computerized system for some forms of duties is a very challenging task, and also a complicated and demanding problem in this modern world. Besides, pattern recognition is the basic component in computer-vision and a framework focused on artificial intelligence.
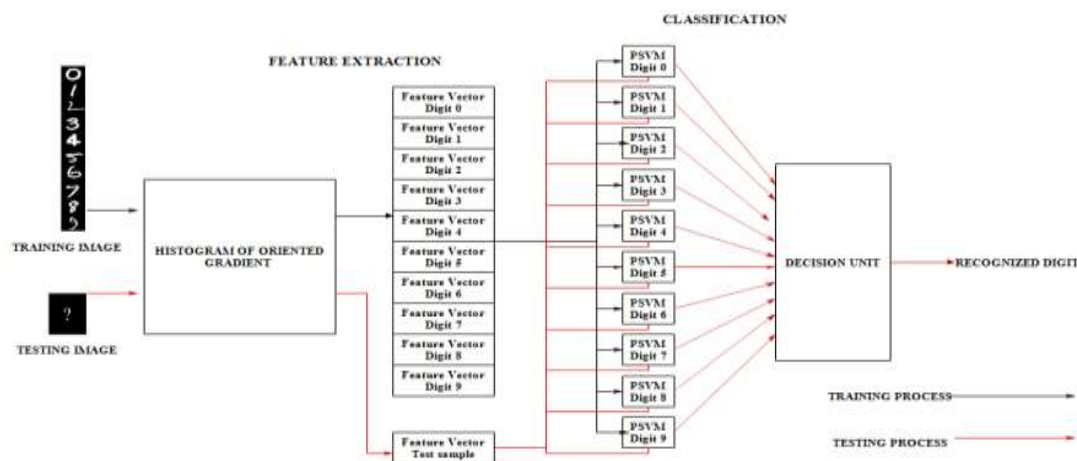

Figure 3: Process of the Recognition system

In this paper, we have introduced machine learning models for the identification of hand-written digits. In this paper, we have introduced CSV datasets which will convert into images first and after that to recognize handwritten digits from 0 to 9. The testing was carried out from the handwritten MNIST database open to the public. We have retrieved from the MNIST database 28,000 digit training images and 14,000 digit test images [4]. The accuracy of 95.88% with test output is our multi-component machine learning models.

2. LITERATURE REVIEWS

[1] - Comparison of precision and time on MNIST datasets Between machine learning and deep learning with respective models that are RFC, KNN, SVM, and Multi-layer CNN. Below Processor measurements, GPU can be beneficial for more precision, shortened preparation and testing time, and GPU may help to obtain parallelism and even improved outcomes. The author achieved a good result in CNN.

[2] - A program that focuses on the functions of the Histogram of Centered Gradient (HOG). As this takes less time and PSVM classifier performance is better than the artificial neural network, the

Proximal support vector machine over the standard SVM classifier was used. 10 class linear PSVM won 98.65 percent of 20,000 samples taken for both preparation and research outcomes for preparation 59 milliseconds (1,000 samples for a digit). The framework has often retained a minimal function vector dimension without including an unnecessary decrease in dimensionality and less training period.

[3] - 98.65 percent accuracy with PSVM and reduced time for the PSVM classifier from each test and test set, 109 seconds (by the ANN) to 59 milliseconds on 10,000 samples, respectively.

[4] - A review that has extended the dataset of MNIST. You've made the latest dataset addressing further topics of grouping. The Restoration EMNIST databases were identified.

[5] - With a gradient descent backpropagation algorithm, a dataset of 5,000 MNIST instances was trained and then tested with a feed-forward algorithm with the number of hidden layers and iterations and the accuracy achieved was 99.32%. 35 neurons and 250 iterations of the Multilayer Perceptron (MLP) neural network were located. 99.32 percent accuracy in training and 100 percent accuracy in training was provided by the proposed method.

[6] - Using a network that employed scattered biologically functioning neurons spike level below 300 Hz ensuring consistency in the classification of MNIST Database 98.17 percent.

[7] - The author revealed that he uses deep neural networks with strong spikes that the weighted spike model proposed hit substantial Latency and number of spikes reduction in a grouping. This leads to more rapidly and more resource-sensitive than the traditional neural spiking network.

[8] - To test the feasibility of the concept the author carried out a comprehensive computer circuit layout co-design to recognize digitally using the manual digits dataset of MNIST. Simulations of equipment to systems implemented by author and demonstrates that the planned skyrmion-based strategies in deep CNNs will accomplish tremendous changes in dynamism usage.

[9] - Deep metric learning developed a hand-written character recognition. The author has produced a new handwritten dataset using the Urdu-Characters model, with classes for profound metrics.

[10] - For IoT applications, The author used a Sparse Deep Neural Network (S-DNN) Processor that measured its high accuracy of classification (98.36% for the MNIST test dataset).

[11] - Used a complex vision sensor Active Perception to identify an NMNIST data collection, with an error rate of 2.4%.

[12] - The authors used Auto-Encoder for MNIST Anomaly Identification for Sparse Representations Learning with variance.

3. PROPOSED METHODOLOGY

3.1. HAND-WRITTEN IMAGES

It is understood that a hand-written dataset is commonly used in computer analysis model evaluations such as machine learning and deep learning. Many model classifiers mostly use the digit groups. Other researchers are however accountable for the alphabet set of groups to display strength and scalability. The research model deals with the description, the fundamental aspects, and algorithm processes, of classification tasks in slightly different ways[5]. Depending upon the number of students the research model is also different. Others vary in training and test breaks, while others perform different image pre-processing processes.
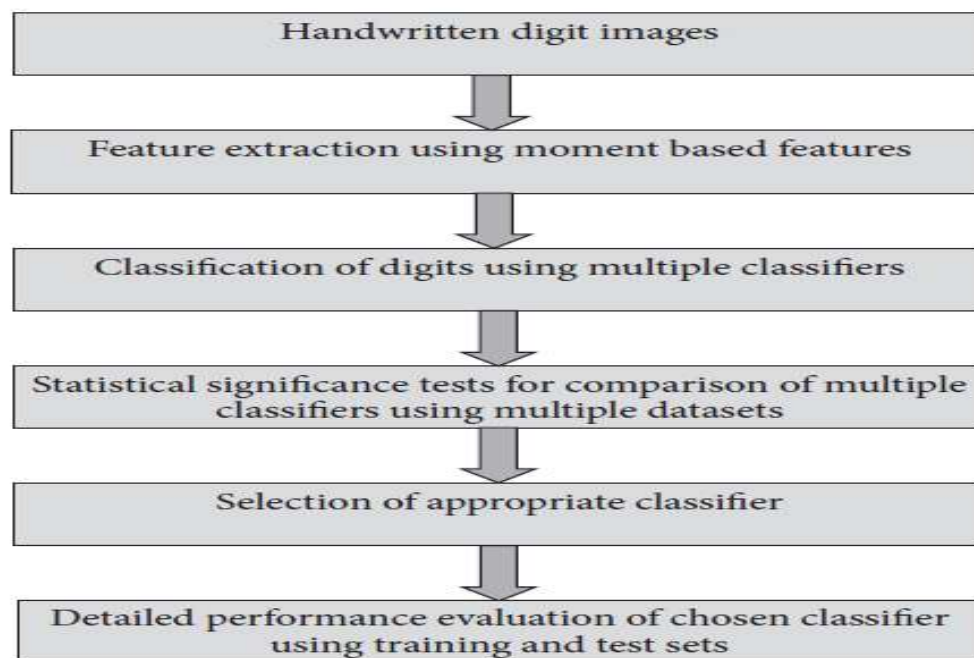
Figure 4: Proposed Methodology Steps

3.2. HAND-WRITTEN DATASETS

The MNIST image datasets are measured from 128 * 128 pixels to 28 * 28 pixels in size decreased. The 282-pixel image representation of the MNIST dataset is 8-bit. First, the grey level picture pre-processed is based on the center mass pixel measurement. In the middle of the 282-pixel images, it is placed to establish the compatible formats of the MNIST dataset. The data collection is available for further study and testing and for further pre-processing. While a greater collection of 814,255 photos is included in the NIST dataset, MNIST takes only a limited part of the study, since the database requires just a ten-digit range, from zero to nine. MNIST data are quite common because of its preparation. The competency of the classification models is used as a standard. Thousands of researchers have used, modified, and checked the dataset that shows that freshly built models can be validated accurately and adequately [13]. Fast access and comprehensive use ease the analysis and dissemination of results among researchers. It mentions several recent research on the usage of MNIST data sets for machine learning.

3.3. IMAGE PRE-PROCESSING

The first step is MNIST dataset is generated and broken down into two different preprocessed datasets. The first preprocessed database is a regular gray-scale, and the second preprocessed dataset is a binary gray-scale [1]. These two methods of preprocessing have been selected because they make it possible to translate the dataset to a low figure while maintaining its aspect ratio [8]. This research paper will perform the tests, and the MNIST dataset was created with two separate formats. The aim of two sets of pre-processed data collections is to monitor the learning performance accuracy of the machine learning model with different pre-processed pictures. This makes scientists consider how machine learning works in diverse pre-process picture formats. The neural network's input format values may depend on how the data set is pre-processed. The models produced are fed with the data sets pre-processed.

3.4. IMAGE SEGMENTATION

Segmentation is a process in the picture which extracts individual characters. Two ways of segmentation occur. They are Segmentation Implicit and Segmentation Explicit. In tacit segmentation, without the method of segmentation, the terms are remembered. Yet terms are expected by the extraction of individual characters in specific segmentation.

3.5. FEATURE EXTRACTION

This is the essential step of the method of recognition, and the recognition algorithm begins from it as well. Each character includes features of its own. It involves a series of laws in which each law defines a behavior's character. In this step, the extraction of certain features is completed.

3.6. MACHINE LEARNING MODELS

(1) Decision Tree Classifier - The classifier for the decision tree is part of supervised learning. Exits in machine learning and other areas, including other distinct supervised learning algorithms. The most used solution to addressing the challenges of classification and regression. The primary objective is to create a training model that predicts the class or targeted values of the rules for learning decisions derived from training and past data. By determining the list of rules of the handwritten digits dataset, the core objective of the model is to create a small tree that can classify the unknown class or example. The DT classifier uses several separate methods for data retrieval, visual identification, predictive text, and machine learning [6]. The research study focuses on the hunt's algorithm to create multiple decision trees based on the C4.5 simulation algorithm of the existing DT. J48 is an open-source platform for the application of the C4.5 algorithm [6], named the Weka DM platform. Second, the algorithm's operating procedure chooses the parameters for the root node and generates a branch for a particular attribute. By assessing uncleanness for each child node, the approach preferred the best split selection from the dataset. Entropy is used to compute the equations for GINI, entropy, and child nodes for the uncleanness degree, usually the C4.5 approach.

<div style="margin-left:2em">

Pseudocode: Decision Tree Classifier
  i.   S1 was a set of classified instances
  ii.   S ≠ Null number of attributes > 0
  iii. The procedure of Decision tree
  iv.   Repeat
  v.   gainMax←0
  vi.   split← null
  vii. e← Entropy of attributes
  viii. Compute for all attributes α in S (store in A)
  ix.   Split(S, new A)
  x.   Till all trees partition
  xi.   End Procedure

</div>

(2) SVM - The SVM or Support Vector Machine is a particular form of supervised learning system intended to distinguish data points in a high-dimensional space by optimizing the margin between classes [3]. SVM is a representation of examples as points in space, mapped by a fair gap that is as comprehensive as possible due to the examples of the separate classes. New

examples are mapped into the same space after that and are supposed to remain in a division depending on which side of the distance they land on [14]. The optimal algorithm is built by a "training" process in which training data is adopted to create an algorithm capable of distinguishing between classes already identified by the operator (e.g. patients vs. controls) and the "testing" process in which the algorithm is adopted to randomly determine the category to which a new experience belongs [3]. It also gives a very reliable representation of classification over the training records and generates ample search room for possible data parameters to be correctly categorized. Therefore, it still promises no less than a rational subset of the data for a variety of parameter combinations. It's often easier to scale the data in SVM; since the performance would be incredibly improved. Therefore, for a huge dataset, be alert, since it can contribute to a rise in training time.

Pseudocode : SVM Classifier
  i.   Dataset of Training
  ii.   Dataset of Testing
  iii. The procedure of Proximal SVM
  iv.   Input Layer
  v.   L SVM classifier
  vi.   Training dataset of S;
  vii. Labels Y
  viii. Repeat and Hyperplane Check
  ix.   Train Classier S* via L
  x.   End repeat
  xi.   L (output)
  xii. End Procedure

(3) Random Forest Classifier - Random forest as an ensemble of un-pruned regression or classification trees, enabled from training data bootstrap samples, implementing a random collection of features in the method of tree imitation [4]. The prediction is rendered by collecting the ensemble's predictions through voting for classification by dominance. It returns the error rate of generalization and is more potent for noise. Still, RF can also endure from the learning curse of an inherently imbalanced data collection for the training set, similar to most classifiers. Because the total error rate is built to minimize, it would appear to rely more on the estimation performance of the majority party, which repeatedly contributes to low precision for the minority class.

Pseudocode: Random Forest Classifier
  i.   Training samples
  ii.   Testing samples
  iii. The procedure of Decision tree
  iv.   Repeat
  v.   Bootstrap Z* the size of N
  vi.   Tb min node n min
  vii. m random variables, p variables
  viii. Split them
  ix.   {Tb}^B
  x.   Till output trees
  xi.   End Procedure

(4) Naïve Bayes - A basic methodology, representing and studying probabilistic information with plain semantics, is given by the Naive Bayes classifier [4]. Since it depends on two major

simplifying assumptions that predictive attributes are conditionally self-reliant given the class, it is considered naïve and insists that no secret attributes impair the system of prediction. It is a probabilistic classifier focused on rigorous and naive independence assumptions based on the Bayes theorem. With various uses of personal email sorting, email spam identification, offensive material identification, document categorization, emotion detection, language detection [4], it is one of the strongest simple text classification approaches. While this method utilizes naive architecture and oversimplified expectations, In many complex real-world problems, Naive Bayes functions well. Although other approaches are also carried out, such as boosted trees, Max Entropy, Supports Vector Machines, random forests, etc., the Naive Bayes classifier is very effective since it is less costly in terms of computing (both memory and CPU) and allows for a small amount of training data. In contrast, the preparation time for Naive Bayes is slightly shorter relative to alternative strategies.

(5) K-Nearest Neighbor (KNN) - A simpler solution for learning purposes is the KNN (K-Nearest Neighbor) algorithm. The non-parametric approach is primarily used for regression and classification [1]. This approach is somehow indicated as a lazy learner strategy. In comparison, KNN's functionality is focused on the feature similarity algorithm. Once fully trained, this model is trained on training samples, then similar samples will be identified in test data[18]. Depending on the K points in the handwritten digit recognition dataset, the KNN working function is set. Therefore, the procedure estimates the digits that are nearest to K, and the plurality votes of the closest points are predicted. A particular class is essentially applied to the main concept of the KNN. And even the nearest neighbor is more intensely described by the individual class.

Pseudocode: K Nearest Neighbor
Procedure of KNN
   i.   K = nearest neighbor and D represent size samples
   ii.   Repeat test sample z=(x$^|$,y$^|$) do,
   iii. Compute the distance - d(x$^|$,x) and (x, y) existence D.
   iv.   Dz subset D the set of K nearest training to z
   v.   End Procedure

4. RESULTS

In this paper after applying different Machine learning algorithms which we have discussed, the maximum accuracy which we were able to achieved was 95.88% by SVM Classifier. The computational time was also reduced. In other Research works the computational time varies with the accuracy and is not stable as we have shown below in the table. It was compared with many different research works and our results are very promising.

Table 1: Accuracy and Time Taken

Machine Learning Algorithms	True-Classified Scenario value (in %)	False-Classified Scenario value (in %)	Time-Taken (in seconds)
Decision Tree Classifier	90.37	9.63	0.53
SVM	95.88	4.12	0.56

Random Forest Classifier	93.85	6.15	0.44
Naïve Bayes	90.83	9.17	3.45
KNN	86.58	13.42	0.55

Table 2: Classification Report (F1 Score, Recall and Precision)

Name of Algorithms	F1 Score	Recall	Precision
Decision Tree Classifier	0.91	0.90	0.90
SVM	0.96	0.95	0.95
Random Forest Classifier	0.94	0.93	0.93
Naïve Bayes	0.87	0.87	0.86
KNN	0.86	0.85	0.84
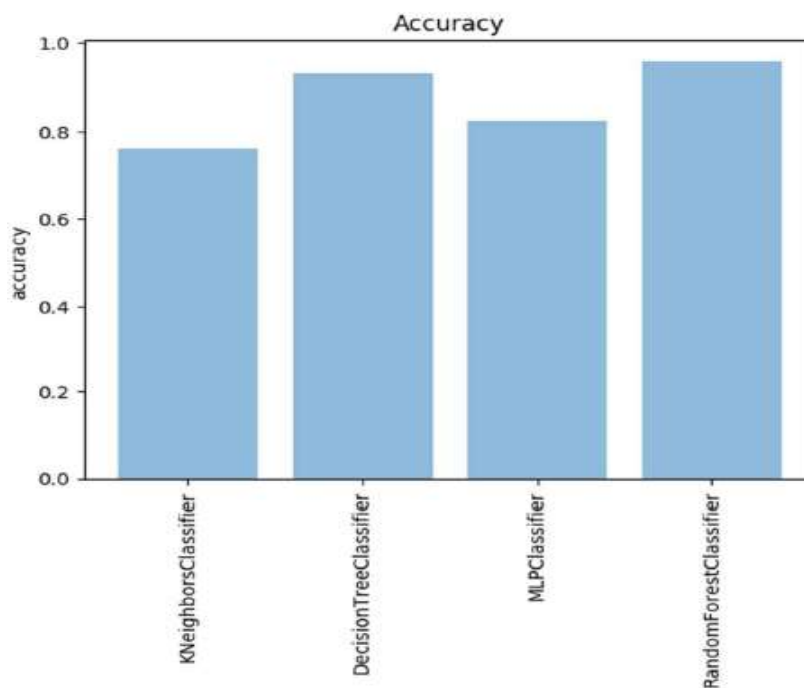
Figure 5: Bar Graph of Accuracy of four ML Models

Dataset Description

Hand-written digital recognition is a broad research area that offers a comprehensive analysis of the industry, including main feature sets, datasets of learning, and algorithms [4]. In comparison to optical character recognition, which focuses on computer-printed display recognition, where specific fonts may be used even for the same scale, script, even glyph properties, the difference between characters is relatively minimal. In offline character recognition system efficiency, the extraction of characteristics and the classification technique play an important role. Various methods for function extraction have been suggested for Method for character identification [1]. Although utilizing techniques such as dynamic programming, neural network, machine learning, and variations of the above techniques, the problems faced in hand-written numeral recognition have been studied [3].

5. CONCLUSION AND FUTURE SCOPE

The key goal of this paper is to find a representation that makes for successful identification of isolated hand-written digits. For the identification of hand-written numerals, numerous machine learning algorithms were used in this paper. The important challenge in every identification method is to resolve the extraction of features and valid classification approaches. In terms of precision and time complexity, the suggested algorithm aims to answer all the variables and well. Among all Machine learning models we got the SVM (Support Vector Machine) recognition method, the total highest accuracy of 95.88 percent is reached. This study is carried out as an initial effort, and the purpose of the paper is to make it simpler to identify hand-written digits without using any common methods for classification.

Future research should allow the usage of a convolution Neural Network architecture which is the topic of deep learning, which provided the best result in the MNIST database and implemented the proposed recognition method by hand. Such more machines may be configured to recognize hand-written characters, identify objects, segment items, recognize handwriting, acknowledge text language, and for potential research, but could also allow hardware deployment for more effective and reliable live results on an online software recognition framework for live test case scenarios.

6. REFERENCE

[1]   M. Z. Alom, P. Sidike, T. M. Taha, and V. K. Asari, "Handwritten Bangla Digit Recognition Using Deep Learning," vol. 6, no. 7, pp. 990–997, 2017.

[2]   S. Majumder, "Handwritten Digit Recognition by Elastic Matching," *J. Comput.*, vol. 4, no. 04, pp. 1067–1074, 2018.

[3]   G. Cheedella, "Critique of Various Algorithms for Handwritten Digit Recognition Using Azure ML Studio," *Glob. J. Comput. Sci. Technol.*, vol. 20, no. 1, pp. 1–5, 2020.

[4]   S. M. Shamim, M. B. A. Miah, A. Sarker, M. Rana, and A. Al Jobair, "Handwritten digit recognition using machine learning algorithms," *Indones. J. Sci. Technol.*, vol. 3, no. 1, pp. 29–39, 2018.

[5]   Y. Lecun, L. Bottou, Y. Bengio, and P. Ha, "LeNet," *Proc. IEEE*, no. November, pp. 1–46, 1998.

[6]   I. Ali, I. Ali, A. Khan, S. A. Raza, B. Hassan, and P. Bhatti, "Sindhi Handwritten-Digits Recognition Using Machine Learning Techniques Sindhi Handwritten-Digits Recognition Using Machine Learning Techniques," *Int. J. Comput. Sci. Netw. Secur.*, vol. 19, no. 5, pp. 195–202, 2019.

[7]   D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural

nets for handwritten digit recognition," *Neural Comput.*, vol. 22, no. 12, pp. 3207–3220, 2010.

[8]   D. N. Kumar and H. Beniwal, "Survey on Handwritten Digit Recognition using Machine Learning," *Int. J. Comput. Sci. Eng.*, vol. 06, no. 05, pp. 96–100, 2018.

[9]   Z. Chen, "Handwritten digits recognition," *Proc. 2009 Int. Conf. Image Process. Comput. Vision, Pattern Recognition, IPCV 2009*, vol. 2, pp. 690–694, 2009.

[10]   P. K. Singh, R. Sarkar, and M. Nasipuri, "A Study of Moment Based Features on Handwritten Digit Recognition," *Appl. Comput. Intell. Soft Comput.*, vol. 2016, pp. 1–17, 2016.

[11]   F. Lauer, C. Y. Suen, and G. Bloch, "A trainable feature extractor for handwritten digit recognition," *Pattern Recognit.*, vol. 40, no. 6, pp. 1816–1824, 2007.

[12]   F. T. Shah and K. Yousaf, "Handwritten Digit Recognition Using Image Processing and Neural Networks," *Lect. Notes Eng. Comput. Sci.*, vol. 2165, no. 1, pp. 648–651, 2007.

[13]   S. Preetha, I. M. Afrid, K. H. P, and S. K. Nishchay, "Machine Learning for Handwriting Recognition," vol. 4523, pp. 93–101.

[14]   B. J. Van Der Zwaag, "Handwritten digit recognition: A neural network demo," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2206 LNCS, no. October 2001, pp. 762–771, 2001.