[ICAETBM-2024]
Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)
doi : https://doi.org/10.32628/IJSRSET

Handwritten Digit Recognition Using Deep Learning

Sonali Patil, Shweta Changan, Ketan Bobade, Yash Darekar, Gauri Mahajan, Dr. Chandrakant Kokane
Nutan Maharashtra Institute Of Engineering And Technology, Talegaon(D), Pune

ABSTRACT

Handwritten digit recognition (HDR) remains a crucial task in various domains, including document processing and human-computer interaction. This paper investigates the application of Convolutional Neural Networks (CNNs) for improved HDR performance. We evaluate the performance of our CNN model on MNIST dataset. Our results demonstrate a significant recognition accuracy showcasing the effectiveness of the proposed CNN architecture.

I. INTRODUCTION

To develop a model that can identify and categorize handwritten numbers. Humans are able to observe and comprehend their visual surroundings thanks to the assistance of their eyes and brains. Giving machines the same sense of perception and processing power as humans is the aim of computer vision[5,6]. In computer vision, many techniques for image recognition have been developed. The goal of our study is to create a model that can more accurately identify and recognize handwritten numbers from their photographs[7,8].

Gaining knowledge and experience with concepts related to Convolutional Neural Networks is the aim of this work. Transcribed digit acknowledgment has long been a controversial topic in the example order community. A neural network does remarkably well in information organizing, according to a few experiments.The primary objective of this work is to create reliable and effective techniques for transcription recognition by analyzing several arrangement models that are already in use[9]. The Convolutional Neural Network (CNN) exhibition is covered in this publication..

The outcomes demonstrate that, without compromising on performance, the CNN classifier created a Neural Network with a far higher computational effectiveness. Handwritten digits can be recognized using Convolutional Neural Networks in Machine Learning. To execute the model, essentially, a few libraries were required, including NumPy, Pandas, TensorFlow, and Keras. This talk will focus on the importance of the convolutional neural network. It was also discussed how datasets are divided into training and test sets. A dataset was used to predict handwritten numbers from 0 to 9. The dataset was cleaned, scaled, and shaped. TensorFlow was used to create a CNN model, which was then trained using the training dataset.

II. PROBLEM STATEMENT:

The field of study has been centered on handwriting recognition for about forty years. This research study looks at the behavior of classification algorithms (CNN) on a large handwriting dataset to forecast a digit. In the construction of these recognition systems, machine-learning techniques are becoming more and more important, particularly when used with neural networks like CNN/ANN. Over the last forty years, handwriting recognition has been the main area of research interest. This research project's objective is to predict a digit by examining how classification algorithms (CNN) behave on a sizable handwriting dataset.. Machine-learning approaches have become increasingly important in the design of these recognition systems, particularly when used with Neural Networks like CNN/ANN.

Our goal is to create a model that uses CNN concepts to recognize and classify handwritten numerals in images. Our work's primary goal is to construct a model for digit identification and classification, but it can also be used to analyze handwritten letters and other documents. Understanding Convolutional Neural Networks and applying them to a system for handwritten number recognition using the created dataset is the main goal of the proposed system.

III. RELATED WORK:

A support vector machine (SVM)-based real-time handwritten digit classification system was created by Ahmed Hafiz (2018), however it is not appropriate for large data sets.

A Handwritten Digits Classification using Back Propagation Network was proposed by Anuj Dutt (2017).The fundamental process of neural network training is backpropagation. This technique involves adjusting a neural network's weights according to the error rate recorded in the preceding epoch.It is less effective, though. by using Keras and Theano as the backend for a convolutional neural network.

In an effort to minimize error rates in handwriting recognition, Denker J.S. (2019) designed Handwritten Digits Recognition. In one study, 3-NN trained and tested on MNIST yielded an error rate of 1.19%. The multimodal neural architecture known as the Coherence Recurrent Convolutional Network (CRCN) is used.It is employed to retrieve sentences from an image. To get around the shortcomings of conventional convolutional layers, some academics are working on developing novel methods. Using MNIST datasets, one strategy that may be used for improved performance is NCFM (No combination of feature maps).

Haider (2020) created a brand-new, difficult Arabic dataset that was gathered from various school study levels. After putting in great effort to distribute and gather digital forms from hundreds of elementary, high school, and college students, a sizable dataset was gathered. He put a lot of effort into creating a difficult Arabic digit dataset after seeing that there were few and undemanding datasets available.

Mukesh N. (2018) carried out a Handwritten Character Classification. Using k-means clustering machine learning algorithms, however, has high baseline error and low accuracy, making it unsuitable for large data sets.

An artificial neural network-based method for handwritten digit recognition was proposed by Nitin Kali Raman (2021).Artificial neurons, which resemble neurons in a biological brain somewhat, are a group of interconnected units or nodes that form the foundation of an ANN. Similar to the synapses in a living brain, every link has the ability to communicate with other neurons. After processing a signal, an artificial neuron can communicate with other neurons that are connected to it.It was a High Baseline Error, though.

A dataset and recognition system for handwritten digits was created by Plamondon, R. (2018) and trained on both convolutional and artificial neural networks. The accuracy evaluation metric was used to determine the average error for both networks. CNN has a lower average error rate than a CPU-based artificial neural network. Nevertheless, CNN training on a CPU required less time than training on an artificial neural network. However, CNN does a superior job in image classification. In summary, the accuracy of recognition increases as the model is trained with CNN; nevertheless, training on a GPU can yield the best results for CNN classification.

A Multilayer Perceptron (MLP) Neural Network was proposed by Saeed Mansoori (2020) to identify and forecast handwritten numbers from 0 to 9.One fully connected type of feedforward artificial neural network

(ANN) is the multilayer perceptron (MLP). The term "MLP" is used in an imprecise manner; it can refer to any feedforward artificial neural network (ANN) or, more precisely, to networks made up of many layers of perceptrons that are activated by threshold.A dataset obtained from MNIST was used for training and testing the suggested neural system.

Gaussian Naive Bayes was used by Shamim S.M. (2018) to introduce machine learning algorithms for handwritten digitizer recognition.A Naive Bayes variation that supports continuous data and adheres to the Gaussian normal distribution is called Gaussian Naive Bayes.This has the low precision as a downside[10].

via the MNIST dataset, Shyam R. (2017) performed Handwritten Digits Classification and found that deep networks perform better when trained via straightforward back-propagation.But in contrast to NORB and CIFAR10, their architecture yields the lowest error rate on MNIST[11].

Sonia Flora (2016) created a support vector machine (SVM)-based handwritten Digits Classification system.Support vector machines evaluate data for regression and classification using supervised learning models and related learning methods.but achieved a lower level of accuracy than convolution neural networks (CNNs)[12].It provides less accuracy because it is a vast dataset.

IV. METHODOLOGY:

Data Acquisition: Getting the MNIST dataset, which is made up of a test set of 10,000 examples and a training set of 60,000 examples, is the first stage. TensorFlow and Keras libraries make it simple to access the dataset.

Data Preprocessing: In order to get the data ready for training, preprocessing procedures must be completed before supplying it to the model. This entails bending the photos to the necessary format, standardizing the pixel values to a range between 0 and 1, and sometimes expanding the dataset to enhance model generalization.Model Architecture Design: The model architecture's design forms the basis of the handwritten character recognition system. We'll use a CNN architecture in this project, which includes of fully connected layers for classification after convolutional layers for feature extraction.

By experimenting and assessing performance, the precise architecture can be altered.

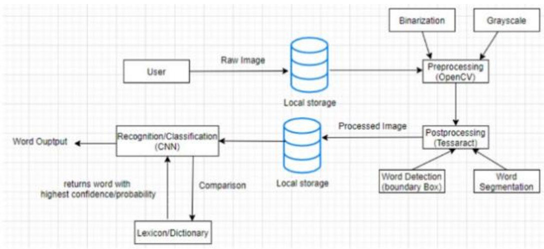
Model Training: After defining the architecture, the model must be trained using the training data that has already been processed. The model gains the ability to link input photos to their corresponding labels—that is, the handwritten digits—during training. In order to minimize the loss function, training entails iteratively changing the model's parameters (weights and biases) using optimization methods like stochastic gradient descent (SGD) or Adam.

Model Evaluation: To determine the model's capacity for generalization, its performance must be assessed on a different test set after it has been trained. Metrics like recall, accuracy, precision, and F1-score are calculated to assess how well the model classifies handwritten letters.

Adjusting Hyperparameters: The performance of the model can be greatly affected by fine-tuning its hyperparameters, which include learning rate, batch size, and dropout rate. In order to determine the ideal configuration, hyperparameter tuning entails methodically modifying these parameters and assessing the model's performance.

Deployment & Integration: The trained model can be integrated and deployed into practical applications once it has reached a sufficient level of performance. This could entail creating an interface for users to interact with the model, accelerating its inference speed for real-time processing, and making sure it works in a variety of contexts and platforms.

V. ARCHITECTURE:



Fig(1) ARCHITECTURE

[1] User: By putting a term into the database, the user starts the procedure.

[2] Local Storage: The word is kept in the local storage.

[3] Preprocessing: OpenCV is used to preprocess the raw image. Usually, preprocessing entails sharpening, noise reduction, and other methods to enhance the quality of the image.

[4] Feature Extraction: The split words are used to extract features. The words are recognized using these characteristics.

[5] Binarization and Grayscale Conversion: After converting the image to grayscale, binarization is performed. Each pixel is changed to either black or white using binarization.

[6] Tessaract postprocessing: Tessaract is an open-source optical character recognition (OCR) engine used to enhance word 171ecognition accuracy.

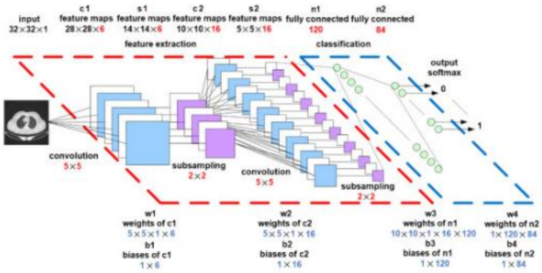
[7] Word Segmentation: Next, each word is separated from the previously processed image. A method known as connected-component analysis is used for this.

[8] Word Detection (Boundary Box): Each recognized word has a bounding box drawn around it.

[9] Recognition/Classification (CNN): To identify the words in the image, a Convolutional Neural Network (CNN) is employed. CNNs are a subset of deep learning neural networks that excel at tasks involving image recognition.

[10] Lexicon/Dictionary: The recognized words are compared to a lexicon or dictionary to ensure they are valid words.

VI. ALGORITHM USED:



Fig(2) CNN ALGORITHM

Because convolutional neural networks (CNNs) can automatically learn spatial information directly from the input image, they have become an effective tool for handwritten digit detection.

Convolutional Layers: These layers extract local information such as forms and edges by applying filters, or kernels, that move across the input image. In a tiny area of the image, each filter finds a certain feature.

To capture different aspects in the image, multiple filters are applied.

Pooling Layers: These layers down sample the feature maps generated by the convolutional layers. This reduces the dimensionality of the data, making the network more computationally efficient and less prone to overfitting. Pooling techniques like max pooling select the highest value in a specific region, capturing the most prominent feature within that area.

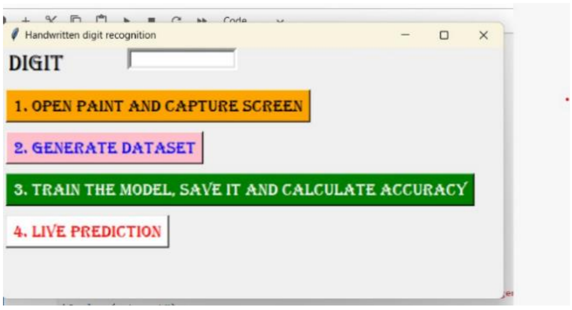
Activation Functions: By adding non-linearity to the network, these functions enable it to understand intricate feature correlations. ReLU and Softmax are popular activation algorithms in CNNs for handwritten digits.

(a) Function of ReLU in Handwritten Digit Recognition: ReLU aids in the introduction of non-linearity in hidden layers of a neural network for digit recognition. This is important since the intricate patterns found in handwritten numerals are beyond the scope of linear models. ReLU enables the network to pick up characteristics that are critical for differentiating between digits, such as edges, curves, and slopes.

(b) Function of Softmax in Handwritten Digit Recognition: The softmax function converts the raw activations from the previous layer into probabilities in the last layer of the digit recognition network.

Fully-Connected Layers: These layers carry out classification in the last stages of the network using the flattened output from the convolutional and pooling layers. With neurons in one layer fully coupled to neurons in the next, these layers function similarly to conventional neural networks.

VII. RESULT:



Fig(3) GUI

1.) Capture Screen:

With the use of the GUI's screen capture functionality, users can upload photographs of handwritten numbers straight into the system. With the help of this capability, real-time digit recognition is possible without requiring external image files.

2.) Dataset Generation:

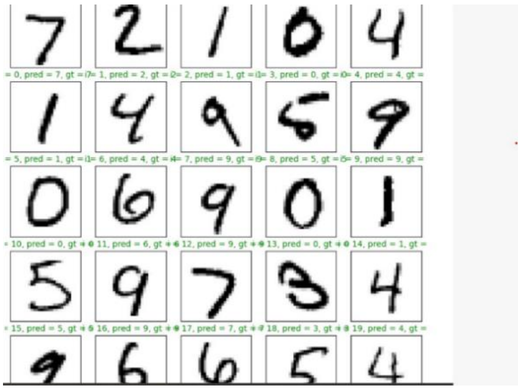
Users can use the tools in the GUI to create a dataset of photographs of handwritten numbers. The digit recognition model uses this dataset for both testing and training. Users can use it to construct custom datasets that are suited to their own requirements and wants.

3.) Model Training and Evaluation:

The generated dataset is used to train a digit recognition model more easily thanks to the GUI. From the UI, users may choose the training algorithm, modify the hyperparameters, and start the training process. Following training, the efficacy of the model in identifying handwritten digits is determined by calculating accuracy measures and evaluating its performance.

4.) Live Prediction:

Live prediction of handwritten digits is made possible by the GUI after the model has been trained and assessed. Digits can be directly drawn by users on the interface, and the trained model makes predictions in real time, showing the identified digit and its confidence level. The performance of the model can be interactively tested and validated thanks to this functionality.



Fig(4)OUTPUT

Pairs of projected digits and the matching ground truth labels are the output from the handwritten digit recognition model, which provide information about the model's performance. Every pair denotes a test instance in which the user's actual written digit and the predicted digit are contrasted.

VIII. CONCLUSION:

In conclusion, a methodical process including data preparation, model architecture design, training, evaluation, and deployment is required to construct a handwritten character recognition system utilizing the MNIST dataset and Python. By employing convolutional neural networks (CNNs) and various strategies like data preparation, hyperparameter tuning, and model optimization, our goal is to develop a dependable and sturdy system that can identify handwritten numbers with exceptional accuracy and dependability.

To effectively create and train the model, we have experimented with a variety of techniques and algorithms throughout the project. We have created a strong basis for testing and assessment by utilizing the MNIST dataset, a popular benchmark for handwritten character recognition. The architecture and hyperparameters of the model have been optimized through repeated refinement and experimentation.
n the end, we are able to utilize the trained model's skills for a variety of activities, from digital assistants and mobile applications to postal automation and document digitalization, thanks to its deployment and integration into real-world applications. We can make sure the system remains relevant and successful in tackling the difficulties associated with handwritten character identification in the current digital era by keeping an eye on it and updating it on a regular basis.

REFERENCES

[1] Samay Pashine, Ritik Dixit, Rishika Kushwah ,"Handwritten Digit Recognition using Machine and Deep Learning Algorithms ", Jun 2021.

[2] Ashutosh Aggarwal ,Karamjeet Singh, kamalpreet Singh,"Use of Gradient Technique for extracting features from Handwritten Gurmukhi Characters and Numerals", 2014, International Conference on Information and Communication Technologies (ICICT 2014).

[3] [1]K. Swetha, [2]Y. Hithaishi, [3]N.L. Tejaswini, [4]P. Parthasaradhi, [5]P.V. Venkateswara Rao, 2021 IJCRT | Volume 9, Issue 6 June 2021 | ISSN: 2320-2882

[4] S. M. Shamim, Mohammad Badrul Alam Miah, Angona Sarker, Masud Rana, Abdullah Al Jobair, "Handwritten Digit Recognition Using Machine Learning Algorithms", Vol. 3 No. 1 (2018): IJoST: April 2018

[5] Kokane, C., Babar, S., & Mahalle, P. (2023, March). An adaptive algorithm for polysemous words in natural language processing. In Proceedings of Third International Conference on Advances in Computer Engineering and Communication Systems: ICACECS 2022 (pp. 163-172). Singapore: Springer Nature Singapore.

[6] Kokane, C. D., Mohadikar, G., Khapekar, S., Jadhao, B., Waykole, T., & Deotare, V. V. (2023). Machine Learning Approach for Intelligent Transport System in IOV-Based Vehicular Network Traffic for Smart Cities. International Journal of Intelligent Systems and Applications in Engineering, 11(11s), 06-16.

[7] Kokane, C., Babar, S., Mahalle, P., & Patil, S. (2022). Word sense disambiguation: A supervised semantic similarity based complex network approach. Int J Intell Syst Appl Eng, 10(1s), 90-94.

[8] Kokane, C.D., Babar, S.D., Mahalle, P.N., Patil, S.P. (2023). Word Sense Disambiguation: Adaptive Word Embedding with Adaptive-Lexical Resource. In: Chaki, N., Roy, N.D., Debnath, P., Saeed, K. (eds) Proceedings of International Conference on Data Analytics and Insights, ICDAI 2023. ICDAI 2023. Lecture Notes in Networks and Systems, vol 727. Springer, Singapore. https://doi.org/10.1007/978-981-99-3878-0_36

[9] Kokane, C. D., & Sachin, D. (2021). Babar, and Parikshit N. Mahalle." Word Sense Disambiguation for Large Documents Using Neural Network Model.". In 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT). IEEE.

[10] Kokane, C. D., & Sachin, D. (2020). Babar, and Parikshit N. Mahalle." An adaptive algorithm for lexical ambiguity in word sense disambiguation.". In Proceeding of First Doctoral Symposium on Natural Computing Research: DSNCR.

[11] Kokane, C.D., Babar, S.D., Mahalle, P.N. (2021). An Adaptive Algorithm for Lexical Ambiguity in

Word Sense Disambiguation. In: Patil, V.H., Dey, N., N. Mahalle, P., Shafi Pathan, M., Kimbahune, V.V. (eds) Proceeding of First Doctoral Symposium on Natural Computing Research. Lecture Notes in Networks and Systems, vol 169. Springer, Singapore. https://doi.org/10.1007/978-981-33-4073-2_11

[12] Kokane, C., Babar, S., Mahalle, P. (2023). An Adaptive Algorithm for Polysemous Words in Natural Language Processing. In: Reddy, A.B., Nagini, S., Balas, V.E., Raju, K.S. (eds) Proceedings of Third International Conference on Advances in Computer Engineering and Communication Systems. Lecture Notes in Networks and Systems, vol 612. Springer, Singapore. https://doi.org/10.1007/978-981-19-9228-5_15

[13] C. D. Kokane, S. D. Babar and P. N. Mahalle, "Word Sense Disambiguation for Large Documents Using Neural Network Model," *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, 2021, pp. 1-5, doi: 10.1109/ICCCNT51525.2021.9580101.