JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

IDENTIFING THE ACCURACY OF HAND WRITTEN DIGITS, USING MACHINE LEARING IN PYTHON

[1]Vasupalli Shyam Kumar, [2]K.Tulasi Krishna Kumar,

[1]MCA 2nd year, [2]Associate Professor,
[1]Master of Computer Application,
[1]Sanketika Vidya Parishad Engineering College, Visakhapatnam, India

ABSTRACT: This paper presents a straight forward neural network approach for recognizing handwritten digits using convolution. Handwritten digit recognition is considered a challenging task for machine learning algorithms such as KNN, SVM, and SOM, due to the unique style of writing. The proposed method utilizes Convolutional Neural Networks (CNNs) [1] to an MNIST dataset [2]of 70,000 digits with 250 distinct forms of handwriting. The results show that the proposed method achieved 98.51% accuracy in predicting real-world handwritten digits with less than 0.1% loss on training with 60,000 digits, while 10,000 digits were held for validation. Overall, this study demonstrates the effectiveness of CNNs in improving handwritten digit recognition performance.

***Key words*: Convolution neural networks, MNIST dataset, TensorFlow, OCR, Segmentation, Cross-Validation**

I.INTRODUCTION

Advancements in the field of computer vision have led to an increased interest in deep neural networks[3],particularly for applications such as OCR (Object Character Recognition) [4]. OCR is a tool that converts printed or documented letters into encoded text by scanning a document and extracting the information to store it in a digital document. There are two major ways of implementing OCR: recognizing patterns in the characters or through segmentation. Handwritten digit recognition (HDR) is a subset of OCR that detects digits and is faster and more flexible in fields such as medicine, banking, student management, and taxation[5].While the human brain[6] can easily interpret sophisticated images, for a computer, an image is a collection of pixels that are a combination of numbers ranging from 0 to 255, such as RGB, B.W., and grayscale[7]. Information can be extracted from these pixel values, but the human eye can segment the image into partitions to interpret shape, size, and colour which are called features. These features are sent to the visual cortex [8]. which the characteristics and maps them to memory for identification. A neural network consists of layers of neurons that process input data and produce output predictions. The first layer, known as the input layer, takes in the data, which is usually represented as a vector of numbers. The output of each neuron in a layer is determined by the weighted sum of the inputs, which is then passed through an activation function. The final layer, called the output layer, has one neuron per class and produces probability values for each class. The number of neurons in the input and output layers depends on the task, while the hidden layers can have any number of neurons and are not specific to the task. The connections between the hidden layers depend on the activations of the previous layers. The activation pattern in one layer affects the pattern in the next layer, with the neuron with the highest activation indicating the network's choice of class. This paper describes the implementation of a convolutional neural network using and sigmoid activation functions to predict handwritten digits from the MNIST dataset. the years various techniques have been developed for recognizing handwritten digits, which is commonly used to test the performance of AI models. Initially, a segmentation-based approach was used, but with the advancements in machine learning, a segmentation [9] less approach was introduced. Despite the implementation changes, the problem still remains open for anyone to solve. One promising approach is the use of the ASSOM technique, as proposed by Bailing Zhang. This technique utilizes the SOM clustering algorithm and autoencoder neural related work. networks in a nonlinear approach, which extracts several features from each digit. An individual ASSOM is constructed for each digit and compared with ten different construction-related errors to minimize misclassification. This method has shown promising results, even with small training samples. Another approach proposed by Saleh Aly, is for recognizing handwritten numerical strings of arbitrary length using SVM and PCA. The major challenge in this approach is word detection [10]. particularly with overlapping characters. Their method uses hybrid PCA called PCA Net for segmentation and SVM for segmentation classification together called PCA-SVM. The experiment shows high efficiency in recognizing unknown handwritten number classification without any segmentation method applied. Yue Yin and Wei Zhang have concluded that out of all neural network implementations, the CNN method is the most valid for OCR-based image classification systems. They state that OCR has become a fundamental technique in the field of computer vision, and if an image classification model performs well in OCR, it can be used for any image classification systems. In the field of computer vision, specifically in image classification tasks such as OCR, Mahmoud M. Abu Ghosh conducted a comparison of CNN, DNN, and DBN approaches. Their analysis found that DNNs (Deep Belief Networks) outperformed other neural networks with an accuracy rate of 98.08%, but fell behind CNNs in terms of execution time. They concluded that any algorithm would only have a 1-2% error rate when recognizing similar digits. A.K. Jain [9] proposed an approximation-based KNN classification algorithm for recognizing handprinted digits. The algorithm performs matching on the deforming edges and dissimilarities of two characters. Their approach works on patterns in a low-dimensional space where the scaling is 2000 times less, resulting in 99.25% accuracy R. Alhajj developed a new approach called the agent-oriented approach for identifying handprinted digits. This approach appoints agents to each character whose sole purpose is to identify hills and valleys, which are black and white regions. The agents' ability to socialize with each other is a key feature compared to other image classification techniques. Overlapping digits are identified based on the intersection of agent paths, known as cut-points. The accuracy of their method is surprisingly high, achieving about 97%. This outperforms many other ANN-based approaches.

II.DIGIT RECOGNITION SYSTEM

Digit recognition system is the working of a machine to train the itself or recognize the human handwritten digits.it is a hard task for the machine because the hand written digits are not prefect and can be made with many different the hand written digit recognition is the solution to the problem which user the image of a digit and the hand and so on the trained model using the test dataset and making predictions on new, unseen data. In summary, while loading the dataset is a necessary step, it can be considered as a prerequisite for the other stages of the proposed method, which include pre-processing, data encoding, model construction, training and validation, and model evaluation and prediction Once all images have been reshaped, we can proceed to split the data into training and testing sets. This is typically done to evaluate the performance of the neural network on unseen data. The training set is used to optimize the weights of the neural network, while the testing set is used to evaluate its performance. The splitting of data into training and testing sets is an important step in machine learning to ensure that the model generalizes well to new, unseen data. The references may provide additional information on this process. The final stage is model evaluation and prediction, which involves evaluating the performance of the trained model using the test dataset and making predictions on new, unseen data. In summary, while loading the dataset is a necessary step, it can be considered as a prerequisite for the other stages of the proposed method, which include pre-processing, data encoding, model construction, training and validation, and model evaluation and prediction.

III.METHODOLOGY

Based on the information provided, it seems that the proposed method in the paper can be divided into several stages, namely pre-processing, data encoding, model construction, training and validation, and model evaluation and prediction. However, since loading the dataset is a necessary step for all these stages, it can be considered as a prerequisite [11] step. The paper uses MNIST as the primary dataset for training the model. MNIST is a well-known dataset consisting of 70,000 handwritten digit images, which are divided into 60,000 training images and 10,000 testing images. The dataset is stored in the IDX file format, which is a binary file format commonly used for storing large multidimensional arrays [12]. The first stage of the proposed method is pre-processing, which involves preparing the dataset for training by performing various operations such as normalization, scaling, and filtering. The pre-processing stage helps to improve the quality of the data and make it suitable for use in training the model. The second stage is data encoding, which involves converting the input data into a suitable format that can be used as input for the machine learning model. The data encoding stage helps to reduce the dimensionality of the data and make it easier for the model to process. The third stage is model construction, which involves designing and building the machine learning model that will be used to perform the classification task. The model construction stage requires expertise in machine learning algorithms and neural network architecture design [13] The fourth stage is training and validation, which involves training the model using the pre-processed and encoded data and evaluating its performance using the validation dataset. This stage is critical in ensuring that the model can generalize well to unseen data. The final stage is model evaluation and prediction, which involves evaluating the performance of the trained model using the test dataset and making predictions on new, unseen data. In summary, while loading the dataset is a necessary step, it can be considered as a prerequisite for the other stages of the proposed method, which include pre-processing, data encoding, model construction, training and validation, and model evaluation and prediction. After prediction, the output digit and its corresponding accuracy score will be displayed on the window. Our model has achieved a high accuracy score, with a prediction rate of almost 90% The paper uses MNIST as the primary dataset for training the model. MNIST is a well-known dataset consisting of 70,000 handwritten digit images, which are divided into 60,000 training images and 10,000 testing images. The dataset is stored in the IDX file format, which is a binary file format commonly used for storing large multidimensional.

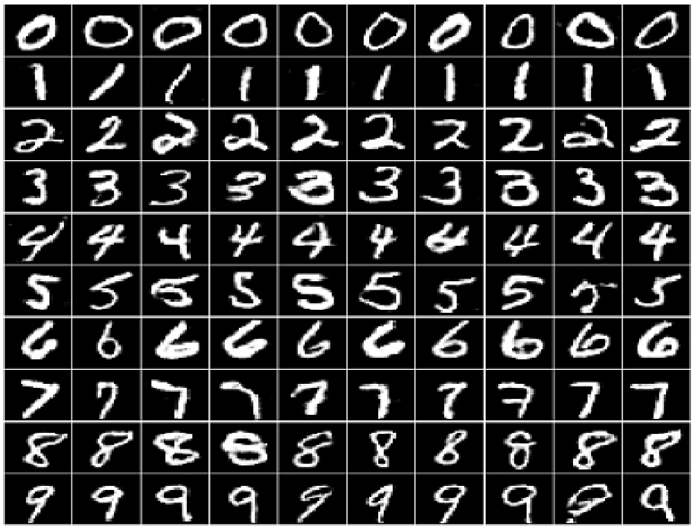

Figure 3: Sample MNIST data

IV.CONVOLUTION NEURAL NETWORK

In simpler world CNN is An article neural network that the specializes is picking our or the detect patterns and make the sense of them , while the testing set is used to evaluate its performance. The splitting of data into training and testing sets is an important step in machine learning to ensure that the model generalizes well to new, unseen data. The references may provide additional information on this process. Once all images have been reshaped, we can proceed to split the data into training and testing sets. This is typically done to evaluate the performance of the neural network on unseen data. The training set is used to optimize the weights of the neural network, while the testing set is used to evaluate its performance. The splitting of data into training and testing sets is an important step in machine learning to ensure that the model generalizes well to new, unseen data. The references may provide additional information on this process.

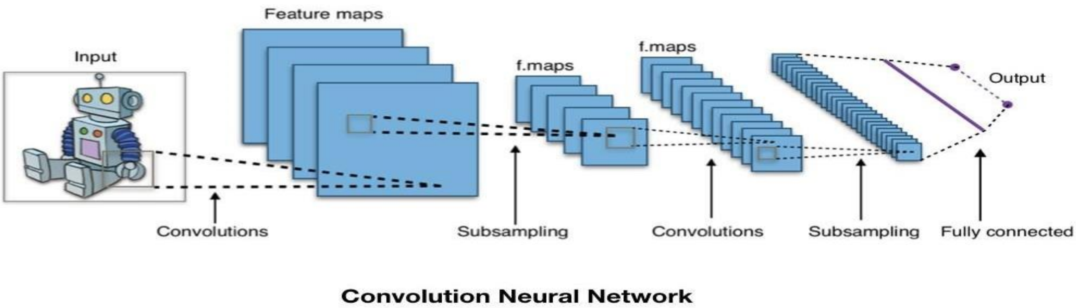

Figure 4: convolution neural network

V.PRE PROCESSING

Once the data is loaded, we typically split it into input and output components. In this case, we have X as the input, which is the image data, and y as the output, which is the corresponding label for each image in X. As illustrated in figure 3, the first layer of our neural network is a convolutional layer. Convolutional layers [14]treat each pixel in the image as a separate neuron in the neural network. Thus, we need to reshape each image[15] such that each pixel value is in its own space. In other words, we convert the original 28x28 matrix of grayscale values . This is typically done to evaluate the performance of the neural network on unseen data. The training set is used to optimize the weights of the neural network, while the testing set is used to evaluate its performance. The splitting of data into training and testing sets is an important step in machine learning to ensure that the model generalizes well to new, unseen data. The references may provide additional information on this process into a 28x28x1 tensor. Once all images have been reshaped, we can proceed to split the data into training and testing sets. This is typically done to evaluate the performance of the neural network on unseen data. The training set is used to optimize the weights of the neural network, while the testing set is used to evaluate its performance. The splitting of data into training and testing sets is an important step in machine learning to ensure that the model generalizes well to new, unseen data. The references may provide additional information on this process.

Figure 5: Flowchart

VI.DATA ENCODING

Since we are utilizing cross-categorical entropy as our loss function, it is necessary to inform the network that the labels provided are categorical in nature [16]. This step is optional.

VII.MODEL CONSTRUCTION

After data encoding, the images and labels are ready to be fitted into our model. Summary of the model can beseen

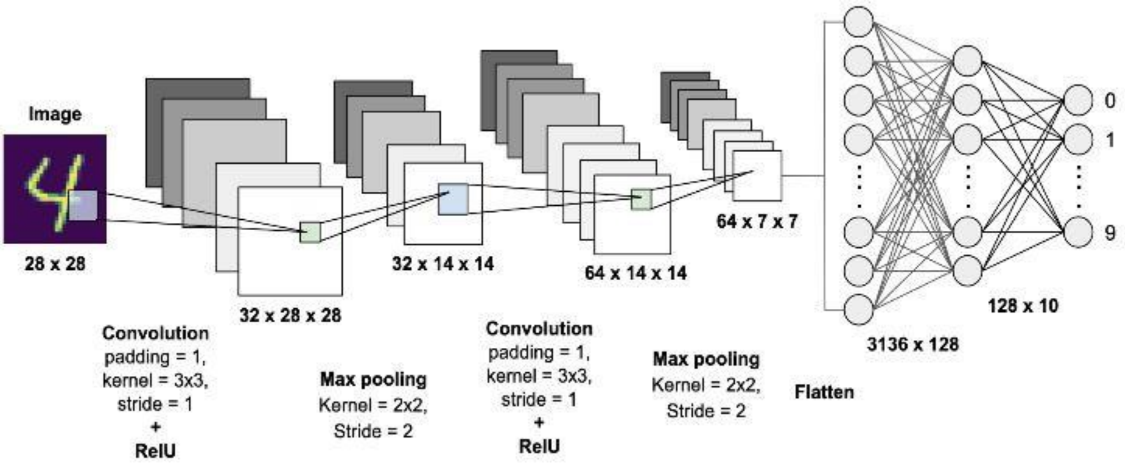

Figure 7: Proposed model

VIII.MODEL SUMMARY

Our model is a binary classification model with a feature extraction step using convolution. We first apply a set of 32 3x3 convolution filters to a 28x28 image, followed by a max-pooling layer [17]with a pooling size of 2x2.We then apply another set of 64 3x3 convolution filters, resulting in 7x7 images. The output layer is a categorical layer with 10 neurons [19]. representing the 10 possible classes of our classification task.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 26, 26, 32)        320
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 13, 13, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 11, 11, 64)        18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 5, 5, 64)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 1600)              0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               204928
_________________________________________________________________
dense_3 (Dense)              (None, 10)                1290
=================================================================
Total params: 225,034
Trainable params: 225,034
Non-trainable params: 0
_________________________________________________________________
```

Figure 8: Model Summary

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. This is typically done to evaluate the performance of the neural network on unseen data. The training set is used to optimize the weights of the neural network, while the testing set is used to evaluate its performance. The splitting of data into training and testing sets is an important step in machine learning to ensure that the model generalizes well to new, unseen data. The references may provide additional information on this process.

IX.K-NN ALGORITHMS

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on SupervisedLearning technique. The one of the simplest learning case is the standard approach compiling we utilized network utilize . SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, Once all images have been reshaped, we can proceed to split the data into training and testing sets. This is typically done to evaluate the performance of the neural network on unseen data. The training set is used to optimize the weights of the neural network, while the testing set is used to evaluate its performance. The splitting of data into training and testing sets is an important step in machine learning to ensure that the model generalizes well to new, unseen data. The references may provide additional information on this process This is typically done to evaluate the performance of the neural network on unseen data. The training set is used to optimize the weights of the neural network, while the testing set is used to evaluate its performance.

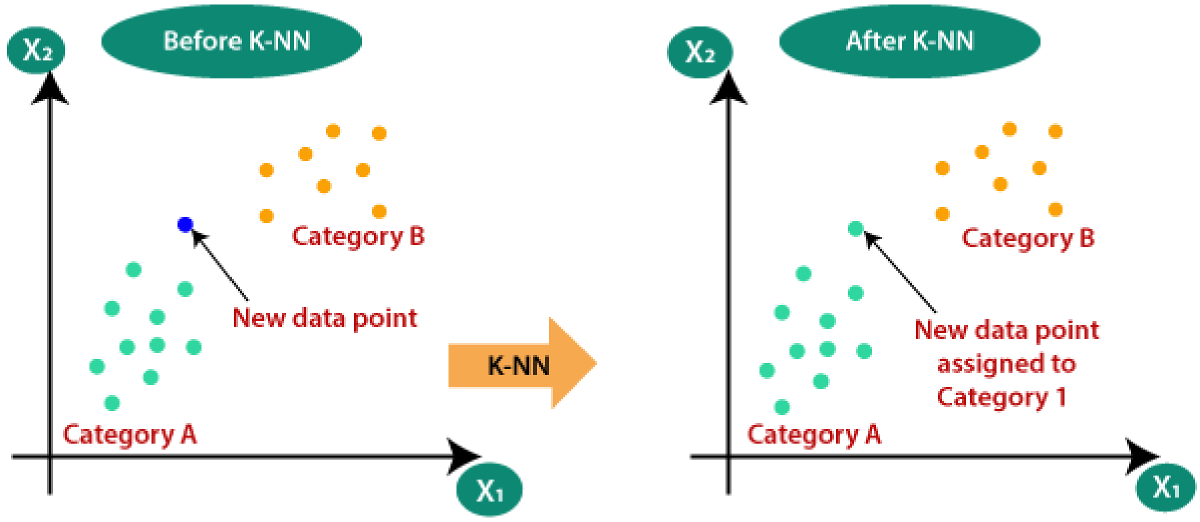

Figure 9: K-NN algorithm

X.SUPPORT VECTOR MACHINE

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine, SVM algorithm.

X.A. LINEAR SVM

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine, SVM algorithm.

X.B. NON - LINEAR SVM

Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. SVM algorithm resizing it to a specific size, normalizing the pixel values[23].and transforming the image into a matrix format. These pre-processing steps enable us to use our model to classify real-world images effectively.

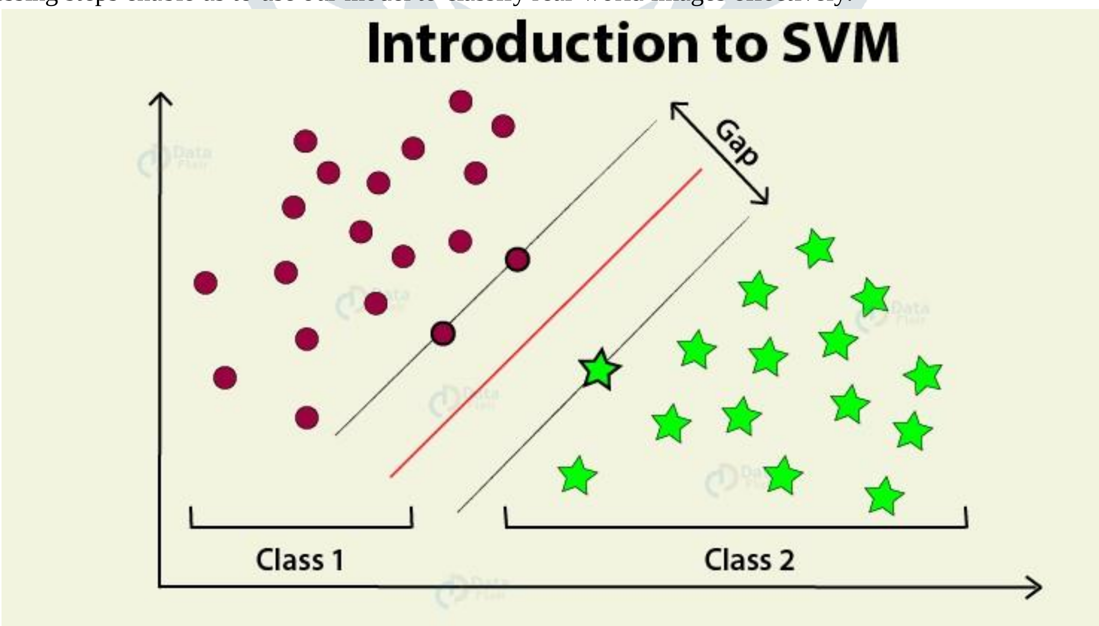

Figure 10: SVM working graph

XI.TRAINING & VALIDATION

We constructed a model , and subsequently, we utilized the standard approach of compiling the model using adam optimizer and cross-entropy loss function for training a convolution neural network. We trained the model for 100 iterations, but since an increase in the number of iterations may lead to overfitting, we stopped the training process when the accuracy reached 98%.

XII.MODEL EVALUATION & PREDICTION

To perform image classification prediction on real-world images, some pre-processing steps are necessary because the model was trained using grayscale raster images. The pre-processing steps involve loading the image, converting it to grayscale, resizing it to 28x28, converting it to a matrix form, and then reshaping the matrix into 28x28x1. Once the image is pre-processed, it is passed through the neural network to predict its label. The neural network output [25] is a list of 10 activation values ranging from 0 to 9, with the highest value corresponding to the predicted label for the image. This process was described in.

XII.RESULTS AND DISCUSSION

The model we have built is intended to be used with real-world data [18].which is quite different from MNIST raster images [24].In order to use our model with real-world images, a significant amount of pre-processing is required to transform a real image into a format that resembles a raster image. This pre-processing may involve steps such as converting the image to grayscale, resizing it to a specific size, normalizing the pixel values [23].and transforming the image into a matrix format. These pre-processing steps enable us to use our model to classify real-world images effectively.

XIV. ACCURACY SCORE

In our training process, the model achieved a training accuracy of 98.21% and a validation accuracy of 98.51% by the end of the second epoch, after which training was stopped. The model also achieved a training loss of 5% and a validation loss of 4%. The accuracy and loss progression [20] can be seen in After pre-processing, we predict the label of the image by passing the pre-processed image through the neural network. The output we get is a list of 10 activation values 0 to 9, respectively. validated the model using a separate test dataset. can be seen in After pre-processing, we predict the label of the image by passing the pre-processed image through the neural network. The output we get is a list of 10

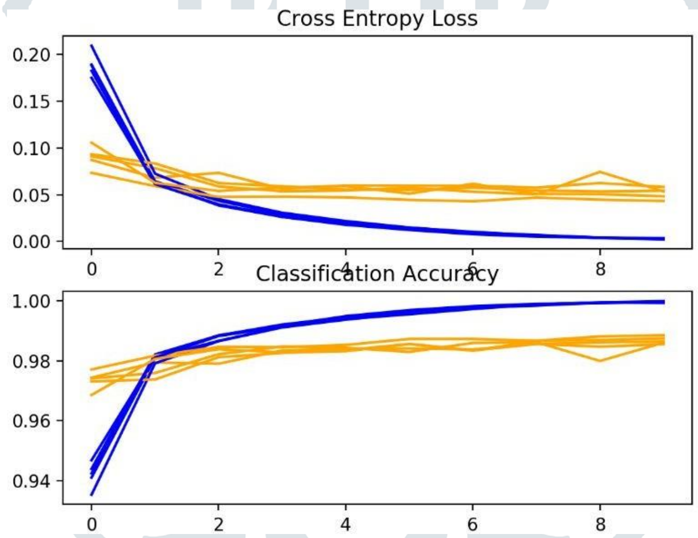

Figure 14: Loss and Accuracy Learning Curves

XV.PREDICTION:

The model we have developed is capable of recognizing both computer-generated and handwritten digits. However, when it comes to accuracy, the model performs better in predicting computer-generated [22] digits compared to real-world digits, as can be seen in Figure.

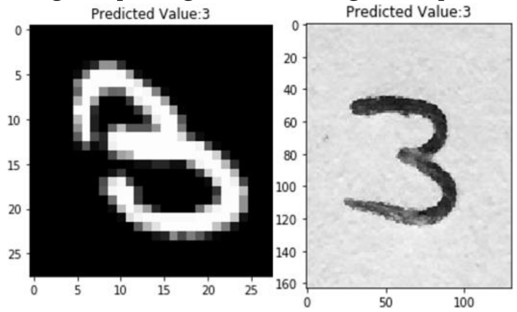

Figure 15: Raster vs Real image prediction

the output digit and its corresponding accuracy score will be displayed on the window. Our model has achieved a high accuracy score. The model's learning rate is highly dependent on the number of dense neurons used and the cross-validation technique employed.

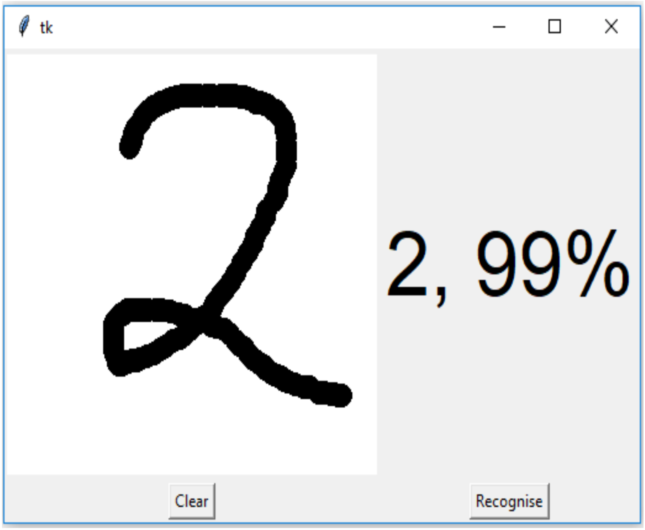

Figure 15.1: Prediction output of 2

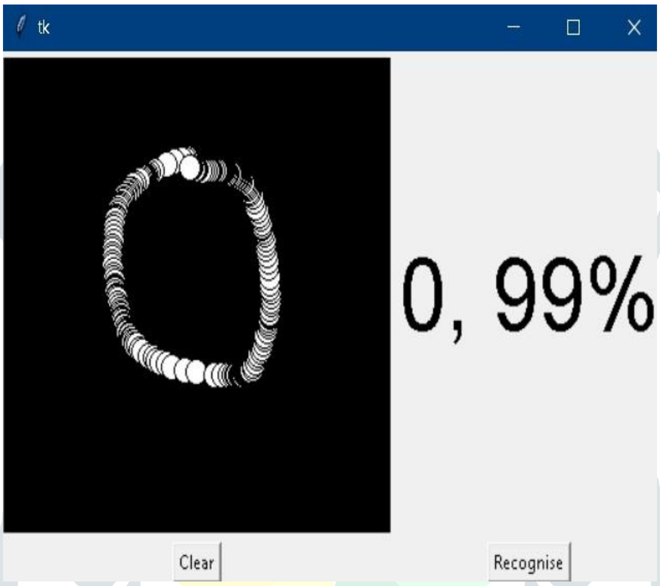

Figure 15.2: Prediction output of 0

XVI.CONCLUSION

The CNN model we used for handwritten digit recognition exhibited outstanding performance, achieving an accuracy of 98%. Moreover, the model was successful in identifying real-world images as well. Both the training and evaluation loss percentages were less than 0.1, which is negligible. The main challenge is dealing with the noise present in real-world images, which requires special attention. The model's learning rate is highly dependent on the number of dense neurons used and the cross-validation technique employed [21].

XVII.REFERENCE

[1] An article reference of Convolutional Neural Networks (CNNs)
https://ieeexplore.ieee.org/abstract/document/8063957
[2] An article reference of MNIST dataset
https://www.sciencedirect.com/science/article/abs/pii/S0262885604000721
[3] An article reference of deep neural networks
https://www.sciencedirect.com/science/article/abs/pii/S1751157718303821
[4] An article reference of Object Character Recognition
https://www.sciencedirect.com/science/article/abs/pii/S0031320319300263
[5] An article reference of taxation
https://www.tandfonline.com/doi/abs/10.1080/01419870.2010.496489
[6]A book reference of human brain
https://journals.sagepub.com/doi/abs/10.1177/1073858402238515?journalCode=nroa
[7] A book reference of grayscale https://link.springer.com/article/10.1007/s11263-010-0324-z
[8] A book reference of the visual cortex
https://link.springer.com/chapter/10.1007/978-1-4757-9625-4_1
[9] A book reference of segmentation
https://link.springer.com/book/10.1007/978-981-10-8818-6
[10] A book reference of word detection https://www.jstor.org/stable/40031892

[11]A web reference of prerequisite https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=97220adb2364063ec43c51ea334cbaf09d94ea12
[12] A web reference of multidimensional arrays https://dl.acm.org/doi/abs/10.1145/276304.276386
[13] A web reference of neural network architecture design https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4221
[14] A web reference of Convolutional layers https://openaccess.thecvf.com/content_iccv_2017/html/Korshunova_Fast_Face_Swap_Using_ICCV_2017_paper.html
[15] A web reference of reshape each image https://pubs.acs.org/doi/abs/10.1021/nl9025665
[16] An article reference of categorical in nature https://www.sciencedirect.com/science/article/abs/pii/S0169023X07001012
[17] An article reference of max-pooling layer https://onlinelibrary.wiley.com/doi/abs/10.1002/ima.22337
[18] An article reference of real-world datahttps://pubs.acs.org/doi/abs/10.1021/es100008x
[19] An article reference of neurons https://www.nature.com/articles/35097581
[20] An article of reference loss progression https://www.sciencedirect.com/science/article/abs/pii/S0140673696082566
[21] An article of reference the cross-validation technique employed https://www.sciencedirect.com/science/article/abs/pii/S221313882030727X
[22] An article of reference predicting computer-generated https://www.sciencedirect.com/science/article/abs/pii/S0278239102156643
[23] A book reference of normalizing the pixel values https://link.springer.com/chapter/10.1007/978-3-319-67558-9_16
[24] A book reference of MNIST raster images https://www.sciencedirect.com/science/article/abs/pii/S2352431620300341
[25] A article refrence The neural network output https://www.sciencedirect.com/science/article/abs/pii/S1352231097004470

BIBILIOGRAPHY

Vasupalli Shyam Kumar is studying his 2nd year Master of Computer Applications in Sanketika Vidya Parishad Engineering College, Visakhapatnam, Andhra Pradesh. With his interest in Python. Machine Learning and as a part of academic project he chose Image processing using Python. The article has been evolved from an idea to understand the flaws in conventional reporting and keeping time consistency, quality report generation in Identifing the Accuracy Of Hand Written Digits, Using Machine Learing In Python.A full-fledged project along with code has been submitted for Andhra University as an Academic Project.

Kandhati Tulasi Krishna Kumar: Project Guide & Training & Placement Officer with decade plus experience in training & placing the students into IT, ITES & Core profiles & trained more than9,000 UG, PG candidates & trained more than 350 faculty through FDPs. Authored various books for the benefit of the diploma, pharmacy, engineering & pure science graduating students. He is a Certified Campus Recruitment Trainer from JNTUA, did his Master of Technology degree in CSE from VTA and in process of his Doctoral research. He is a professional in Pro-E, CNC certified by CITD He is recognized as an editorial member of IJIT (International Journal for Information Technology & member in IAAC, IEEE, MISTE, IAENG, ISOC, ISQEM, and SDIWC. He published articles in various international journals on Databases, Software Engineering, Human Resource Management and Campus Recruitment & Training.