

Handwritten Digit Recognition System using Convolutional Neural Network (CNN)

Goh Kiat Yong¹, Aimi Syammi Binti Ab Ghafar^{1*}

¹Department of Electrical Engineering Technology, Faculty of Engineering Technology,
University Tun Hussein Onn Malaysia, 84600, Pagoh, Johor, MALAYSIA

*Corresponding Author Designation

DOI: <https://doi.org/10.30880/peat.2021.02.01.057>

Received 13 January 2021; Accepted 01 March 2021; Available online 25 June 2021

Abstract: Nowadays our life is inseparable from the electronic data. One of the technically important problems in pattern recognition systems is the handwritten digit recognition. The digit recognition applications include postal mail sorting, bank check processing, data entry forms, etc. The users are emphasizing the accuracy of the system. On the market, there are a lot of handwritten digit recognition system with different algorithm. The basic purpose of the study was to implement an algorithm based on machine learning and improve the CNN architecture (Second Model) to increase the accuracy of handwritten digit recognition system., and to analyses the performance of the proposed algorithm with test data set. There are several machine learning algorithms such as Support Vector Machine, Random Forest, Multilayer Perceptron, Convolutional Neural Network etc. In this project is aimed to use Convolutional Neural Network to complete the task. The MNIST dataset also used in this project. Even though the goal of the project is creating a model to recognize digits, it can be extended for letters in future work with the same concept. The result shows the accuracy of the model is 98.00 % and above after evaluated the model, it was a good result compare other algorithms. Model improvement is done by increasing the depth of the model. The performance of updated version model has a small improvement of accuracy from 98.66 % to 98.96 %.

Keywords: Handwritten Digit Recognition, Convolutional Neural Network, Machine Learning.

1. Introduction

Handwritten recognition is quite common research field nowadays. There are multiple daily task such as bank account number, bank check processing, handwritten data type and invoice analysis, and postal mail reading using handwritten recognition, because of this it become an important area for study[1]. Handwritten digit identification is a difficult issue because of the various characteristics and types of handwriting. Various methods have been study in handwritten digit recognition, such as support

*Corresponding author: aimi@uthm.edu.my

2021 UTHM Publisher. All rights reserved.

publisher.uthm.edu.my/periodicals/index.php/peat

vector machine classifier, deep learning-based classification algorithms, artificial neural network, and so on, but misrecognition of some digits is still unavoidable [2].

With the development of science and technology, handwritten recognition is widely used in electronic devices. There are few examples of notable products in handwritten recognition, Microsoft Tablet PC, WritePad, HWPen, CellWriter, MyScriptStylus, and so on. From the point of view of Human-Computer Interaction (HCI), natural input technology can be used to describe the handwriting [3]. One of the studies shows that if the failure recognition rate is more than 3.00 % is not tolerable by the end users [4].

Nowadays our life is inseparable from the electronic data. In most areas, their work related with electronic data, therefore the handwritten digit recognition is indispensable. There is still quite a lot of area that needs the process of handwritten digit data to convert to electronic data form. The users are emphasizing the accuracy of the system. On the market, there are a lot of handwritten digit recognition systems with different algorithms. The aim of this project is to implement Convolutional Neural Network (CNN) architecture (First Model) for handwritten digit recognition system, improve the CNN architecture (Second Model) to increase the accuracy of handwritten digit recognition system, and analyze the performance of the proposed CNN architecture with test data set. While the goal is to build a model that can recognize the digits, it can be applied to letters and the handwriting of a person. The main objective of the proposed system is to understand and apply the Convolution Neural Network to the handwritten recognition system.

The scope of this project is to create a classification algorithm using Python programming. Besides this, MNIST dataset needs to be used to perform the algorithm. The architecture of CNN for handwritten digit recognition system needs to be studied and understood. The CNN algorithm is chosen due to its accuracy performance compared to other algorithms, and it is also the algorithm commonly used in the market. In my opinion, Python programming is the most user-friendly programming language compared to others, and the MNIST dataset is the most complete dataset for handwritten digit recognition system use.

2. Literature Review

2.1 MNIST dataset

MNIST stands for modified National Institute of Standards and Technology. It was the subset of NIST dataset [5]. The MNIST database consists of a large number of handwritten digits. This data set is generally used to test image processing systems. As a result of the number of data points in the data set and the simplicity of the data set, this data set is used as the hello world in the field of computer vision. A HelloWorld program is a general term used to describe the introductory program in a domain that demonstrates the basic principles of the domain. This dataset consists of handwritten digits as grayscale images where each image is of 28 by 28 pixels. The images have been normalized to fit at the center of the bounding box. The dataset consists of a set of 60,000 images in the training data set and a set of 10,000 images as testing data set. The extended MNIST data set consists of 240,000 training images and 40,000 testing images. The task is to train a model to predict the label of a handwritten digit from the data set, as is evident there are ten possible labels zero to nine, the output of the model is a set of confidence levels for each digit as shown in the figure. Our goal is to train the model such that we can predict the digits with a high level of confidence.

2.2 Machine Learning

An algorithm is needed to solve the computer problem. To transform the input to output sequence of instructions is needed, this is also known as an algorithm. For some of the tasks that do not have any algorithm that can be used, machine learning is playing an important role in this situation. All the data has its pattern, to detect certain trends or regularities of the data is the niche of machine learning [6]. A computer

program is assigned to perform certain tasks in Machine Learning, and it is said that if its observable output in these tasks increases as it gains more and more experience in the performance of these tasks, the system has benefited from its experience. Explain briefly, the machine used the data to do the decisions and predictions. The classifications for machine learning approaches sometimes separated to four type of categories, which is “supervised learning”, “unsupervised learning”, “semi supervised learning”, and “reinforcement learning”[7].

2.3 Deep learning

One of the subfields of machine learning is deep learning. Deep learning enables a multiple processing layer computational model to learn data representation with multiple level of abstraction. These techniques have greatly improved the state of the art of voice recognition, visual object recognition, target identification and many other areas, such as drug discovery and genomics. In order to explore complex structures in large data sets, Deep Learning uses a backpropagation algorithm to show a computer how it can change its internal parameters (computing the representation of each layer from the previous layer representation). The images, video, speech and audio processing by using deep convolutional networks produced breakthroughs, while recurrent networks inspire sequential data, such as text and speech[8].

2.4 Convolution Neural Network

Convolution Neural Networks (CNN) is one of the architectures in deep learning. CNN can draw out the properties from input data, which is multi-layer neural feed-forward network. Neural network back-propagation algorithm is used to train the CNN. Even though, a very large number of data is difficult to learn complex, high-dimensional, non-linear mappings, but CNN has the potential to do it[9]. Compare to traditional image classification algorithm, CNN need only little pre-processing. This state that the CNN learn the filters which have been hand-engineered in conventional algorithms[10]. CNN can auto extracts the invariant salient features and also shift and shape distortions of the input characters at certain degree, this was one of the advantages of CNN[11].

The Convolution Neural Network's role is to reduce the images into an easier to-process type, without missing features that are crucial to making a successful prediction. All convolution neural network is combined of multiple layers, the three main types of layers are convolutional, pooling, and fully connected. Convolution is the first layer where features are collected from an input image. Convolution used small squares of input data to learn image features to preserves the relation between pixels. It is a mathematical operation that involves two inputs, like an image matrix and a kernel or filter. The Section Pooling Layers will reduce the number of parameters if the images are too huge. Also known as subsampling or down sampling, spatial pooling decreases the dimensionality of each map but retains important data. In a fully connected layer, the matrix is flattened into a vector and fed like a neural network into a fully connected layer. The map matrix of features will be transformed to a vector. In order to create a model with completely connected layers, these features will be combined together[12].

2.5 Related work

This section presents a review of recent techniques developed for the handwritten digit recognition along with the advantage and limitation of each individual technique. E. Tuba, M. Tuba, and D. Simian [13], has investigated a new algorithm for handwritten recognition. The goal of this paper is to use a simple feature set as input to support vector machine used for classification. The authors have said that from recent swarm intelligence algorithm, bat algorithm is optimal support vector machine models. It has been modified and used for support vector machine parameter tuning. Their used MNIST dataset to test their method, the error is only 4.60 %. Authors has made a comparison with another two paper which using the same algorithm, their method provide lower error rate with rather simple feature set.

The authors evaluate that this algorithm is robust, it can further improve by using more complex features. Many repositories, for example USPS, can be used to verify additionally.

A. W. Chen [14], has proposed a review on using logistic regression for automatic digit recognition, a popular tool for efficiency improvements in many areas. In this paper, author only train and test two set of digits, which is 0 and 1, 3 and 5. Stochastic gradient descent model is used. Authors vary the training and testing data sample size to examine the model accuracy. Result show that the accuracy remains consistent across all sample size. Author state that, this may be due to the comprehensiveness of the data set, which contains a wide variety of handwritten digit for used. After the experiment author found that the accuracies for 3 and 5 data set is lower than another data set. When the large samples used, the accuracies are higher than the small samples used, small sample is 30.00 % or lower the data available. The accuracies are more than or equal to 85.00 % even in the worst case, this is due to the large variety of handwritten digits comprehensive set (MNIST dataset).

S. M. Shamim, M. B. A. Miah, A. Sarker, M. Rana, and A. Al Jobair [15], has investigate a few different techniques of machine learning for off-line handwritten digit recognition. This paper objective is to ensure the approaches which is effective and reliable. Authors used WEKA to examine the few type of machine learning algorithm, which is Multilayer Perceptron, Support Vector Machine, Naïve Bayes, Bayes Net, Random Forest, J48 and Random Tree. In this paper, the suggested algorithm tries to address both the variables and well in term of accuracy and time complexity. From the observe of the authors, the overall highest accuracy is 90.37 %, 12 which is achieved by Multilayer Perception algorithm. The authors have mention that, this paper is without using any standard classification techniques for handwritten digit recognition facilitate, and this work is done as a first attempt.

C. Ma and H. Zhang [16], has proposes a specific multi-feature extraction and deep analysis approach for handwritten digit recognition. First to remove negative information and preserve relevant features, authors normalize various size and stroke thickness of image in pre-processing. After that, authors suggest unique descriptions of features, including structure features, distribution features, and projection features, since they assume that recognition of handwritten digital images is different from recognition of conventional image semantics. In addition, authors fuse several features into the deep neural networks for semantic recognition. In this paper, extensive empirical studies on the MNIST dataset demonstrate their method's usefulness and supremacy. Authors compare their method with another two method which is self-organizing map (SOM) method and the P-SVM method. Their method has the highest accuracy compare to another two method, which is 94.20 %.

J. Qiao, G. Wang, W. Li, and M. Chen [2], to enhance precision and shorten running time for handwritten digit recognition, an adaptive deep Q-learning technique has been investigated. In adaptive deep Q-learning strategy, to create an adaptive Q-learning deep belief network (Q-ADBN), which will combine deep learning features extracting capability and reinforcement learning decision-making. Firstly, Q-ADBN used an adaptive deep auto-encoder (ADAE) to extracts the original image features, and the extracted features are known as the current states of Q-learning algorithm. After that, Q-ADBN collects Q-function (reward signal) when the current state recognition, and the final handwritten digit recognition is applied by optimizing the Q-function using Q-learning algorithm. The authors have state that the results from the experiment, compare to other similar method Q-ADBN has better accuracy (99.18 %) and shorter running time.

In [17], M. A. Hossain and M. M. Ali has investigate a model that will be able to recognize the handwritten digit from its image with better accuracy. The authors used the Convolution Neural Network concepts and MNIST dataset to complete this experiment. Authors also demonstrate how MatConvNet can be used with CPU training to execute their model and less training time, too. After the experiment, authors get a 99.15% accuracy. Authors state that the more data in the training set, the smaller the impact of training error and test error, and ultimately the accuracy can be improved. 13

Convolution Neural Network is better than other classifiers, and the more the number of hidden neurons and convolution layers, the result can be more accurate, the authors have said.

2.6 Comparison table of existing handwritten digit recognition

Table 1: Comparison of existing handwritten digit recognition

Article	Summary	Keywords	Approach	Dataset
E. Tuba, M. Tuba, and D. Simian, "Handwritten Digit Recognition by Support Vector Machine Optimized by Bat Algorithm," <i>Conf. Comput. Graph. Vis. Comput. Vis.</i> , vol. 4617, pp. 369–376, 2016.	In this paper, determine optimal support vector machine, which involves soft margin tuning and kernel function parameters. Author adjust recent swarm intelligence bat algorithm. 4 histogram projection is used in this experiment. The experiment test the approach on MNIST dataset, compare with other approach, proposed algorithm get a better accuracy.	Swarm intelligence Parameter tuning Handwritten digit recognition	Support vector machine Bat algorithm	MNIST
A. W. Chen, "Review Article Handwriting Recognition and Prediction Using Stochastic Logistic Regression," vol. 05, pp. 5526–5527, 2018.	In this paper, authors proposed logistic regression for automatic digit recognition. Only two pair of digits is used in the experiment. Stochastic gradient descent model is used. Author vary the training and testing data sample size to examine the model accuracy. Result show that the accuracy remains consistent across all sample size. The large samples used, the accuracies are higher.	Machine Learning Digit Identification Classification Models	Stochastic Gradient Descent,	MNIST
S. M. Shamim, M. B. A. Miah, A. Sarker, M. Rana, and A. Al Jobair, "Handwritten digit recognition using machine learning algorithms," <i>Indones. J. Sci. Technol.</i> , vol. 3,	In this paper, authors have presented few type approaches to off-line handwritten digit recognition. The objective of this paper is to ensure effective and reliable approaches for recognition of handwritten digits. Multilayer Perceptron, Support Vector Machine, Naïve Bayes, Bayes Net,	pattern recognition handwritten recognition digit recognition off-line handwritten recognition	Multilayer Perceptron Support Vector Machine Naïve Bayes Bayes Net Random Forest	WEKA

<p>no. 1, pp. 29–39, 2018, doi: 10.17509/ijost.v3i1.10795.</p>	<p>Random Forest, J48 and Random Tree, those approach are show in this paper. The result show multilayer perception has a highest accuracy compare to other. WEKA is used in this experiment.</p>	<p>neural network</p>	<p>J48 Random Tree</p>	
<p>C. Ma and H. Zhang, “Effective handwritten digit recognition based on multi-feature extraction and deep analysis,” <i>2015 12th Int. Conf. Fuzzy Syst. Knowl. Discov. FSKD 2015</i>, pp. 297–301, 2016, doi: 10.1109/FSKD.2015.7381957.</p>	<p>In this paper, authors investigate an approach based on specific multi-feature extraction and deep analysis for handwritten digit recognition. Extensive empirical studies on the MNIST dataset demonstrate their method's usefulness and supremacy. Proposed method has the highest accuracy compare to another two method in this paper shown.</p>	<p>handwritten digit image recognition deep learning</p>	<p>multi-feature</p>	<p>MNIST</p>
<p>J. Qiao, G. Wang, W. Li, and M. Chen, “An adaptive deep Q-learning strategy for handwritten digit recognition,” <i>Neural Networks</i>, vol. 107, pp. 61–71, 2018, doi: 10.1016/j.neunet.2018.02.010.</p>	<p>In this paper, authors proposed an adaptive deep Q-learning strategy to improve accuracy and shorten running time for handwritten digit recognition. An adaptive Q-learning deep belief network (Q-ADBN), which is combination of deep learning features extracting capability and reinforcement learning decision-making. The result of the experiment show that proposed method has better accuracy and running time compare to other.</p>	<p>Handwritten digits recognition Deep learning Reinforcement learning Adaptive Q-learning deep belief network Adaptive deep auto-encoder</p>	<p>Deep learning Reinforcement learning Adaptive Q-learning deep belief network</p>	<p>MNIST</p>
<p>M. A. Hossain and M. M. Ali, “Recognition of Handwritten Digit</p>	<p>The main objective of this paper is to create a better model for handwritten digit recognition. Convolution Neural</p>	<p>MatConvNet ReLU softmax</p>	<p>convolutional neural network</p>	<p>MNIST</p>

using Convolutional Neural Network (CNN),” <i>Glob. J. Comput. Sci. Technol.</i> , vol. 19, no. 2, pp. 27– 33, 2019, doi: 10.34257/gjcstdvol19i s2pg27.	Network and MNIST dataset is used. Authors has demonstrated MatConvNet can be used to implement our model with CPU training. Authors has state that Convolution Neural Network provide better performance than other classifiers.
--	---

From the literature review about the handwritten digit recognition above, it shows that neural network is the trend of image recognition field. In most of the research when comparing various approach, the approach using neural network algorithm will get a better performance compare to other. Deep learning has become the dominant approach for much ongoing work in the field of machine learning.

3. Methodology

There are 4 phases to be considered to achieve the goal of this project, as shown in Figure 1. Phase 1 is the acquire and prepare data. Next phase is building and compiling of the model. The third phase is training and evaluating the model. The last phase is predicting the output. The first phase is acquired and prepare data. The purpose of this phase is to import the data set and clean the data set. Phase 2 is building and compiling of the model. In this phase choose an appropriate machine learning algorithm for the problem statement and the data at hand. Then choose a specific algorithm within the category by considering various factors such as training time, prediction time, memory requirements, ease of deployment etc. Phase 3 is training and evaluate the model. In this phase the model is train by using the data from the previous phase. The testing data set is used to evaluate the model. The testing dataset representative of the real-world data. The estimate of how well the model will perform on the data the model has not seen during training can be obtained. Phase 4 is predicting the output. In the last phase, the model is used to predict the outputs for the given data. For example, a model is train to recognize handwritten digits using the dataset. Then an own input can be provided by writing a digit and obtain the prediction by using the model.

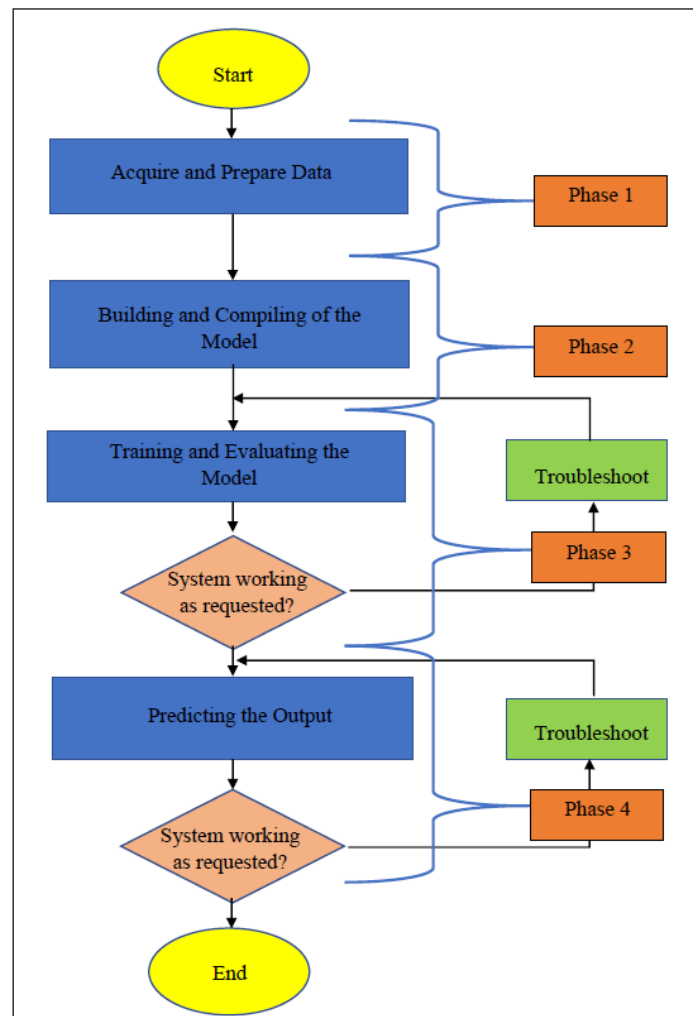


Figure 1: Flowchart of project development

3.1 Programming tools

We will use python as the programming language. Python has been the primary choice for using machine learning because it is beginner friendly and easy to read. Because of its high-level nature, the time to get a working program is reduced significantly compared to languages such as Java or C++.

Anaconda is a popular data science platform. The purpose of using this platform is that if we install anaconda, all the essential libraries are provided out of the box without the need to use the command line to install and manage the packages. Anaconda is a beginner friendly tool that allows the users to get started with data science and machine learning without getting with hung up on installation and configuration of the necessary tools.

The Jupyter Notebook is an open-source web application that allows documents containing live code, equations, visualizations and narrative text to be created and shared. Jupyter Notebook can be used for performing various mathematical operations like numerical simulation, statistical modelling, data visualization etc. using simple commands.

3.2 Data handling

NumPy is a Python math library and a package for general purpose array-processing. It offers a multidimensional array object with high performance and tools for working with these arrays. It also offers mathematical functions at high levels.

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is easy to generate a variety of graphs by using Matplotlib.

3.3 Machine learning libraries

TensorFlow is an open-source library developed at Google. It provides extensive functionality to implement deep learning algorithms. It also provides the functionality to run the training on GPUs or multi core CPUs to reduce the training time.

Keras is a high-level API to abstract away the details of TensorFlow. Keras allows us to quickly implement an artificial neural network but, because it hides the details, it lacks the advanced features that TensorFlow provides.

3.4 Architecture of the convolutional neural network

When talk about the deep learning with neural network, CNN is the better choice for image classification. Convolution Neural Network is different from others algorithm. From the minimal pre-processing pixel image, CNN can recognize it pattern, and it also one of the advantages compare to others algorithm. Most of the CNN architectures adopt the same basic design rules, first adding convolutional layers to the input data, after that down sampling the spatial dimensions (Max pooling) while increasing the feature maps. Third, fully connected layers. Among the CNN operation, these three layers is frequently mention in a lot of CNN research, because there are most import ones in CNN architecture.

3.4.1 Convolution layer

This layer constitutes the fundamental unit of a CNN in which most of calculation are involved. The neurons are arranged in the feature maps, convolution layer is a set of feature maps. The layer's parameters consist of a set of learnable filters (or kernels), which have a small receptive field, but extend through the full depth of the input volume. These feature map filters are convolved to generate a separate 2-D activation map, the output volume is produced when those stacked together along the depth dimension. Neurons shares the weight when there are in the same feature maps accordingly to keep the number parameters low, and the complexity of the network will be reduce [18]. The spatial extend of sparse connectivity between the neurons of two layers is a hyperparameter called receptive field. Receptive field a hyperparameter that sparse connectivity of the spatial extend between the neurons of two layers. Three hyperparameters depth, stride, and zero-padding is used to control the size of the output volume of the convolutional layer. The depth for the number of filters per sheet, the phase for the motion of the filter, and the zero-padding to control the output spatial size. With Backpropagation, the CNN is educated, and the backward pass also includes convolution operation but with spatially flipped filters. Computing dot products with corresponding receptive area elements (of R, G, B channels) and filters are involved in the convolution process. The receptive field window slides through the image, computing internal products spatially and resulting in a map of functions.[19].

3.4.2 Pooling layer

Pooling layer function is to reduce the spatial dimension of the activation maps without losing the information or feature and the number of parameters in the net, thereby to speed up the calculation, as well as to make some of the features it detects a bit more robust. It helps in overfitting problem. Pooling can compute an average or a max. In average pooling the average value is taken from each cluster of the neurons at the previous layer. While max pooling is taking the maximum value from each cluster of the neurons at the previous layer. Max pooling is more common used compare to average pooling nowadays.

3.4.3 Fully connected layer

In Fully Connected Layer, the neurons in this layer are fully connected to all the neurons in the previous layer, as seen in regular neural network. Fully connected layer only is a feed forward neural network. Therefore the last few layer is form by the fully connected layer. The input to the fully connected layer is the output from the final Pooling or Convolutional Layer, which is flattened (unroll all its values into a vector) and then fed into the fully connected layer. The neurons are not spatially arranged (one dimensional) so there cannot be a conv layer after a fully connected layer[19].

3.4.4 Loss layer

The last fully connected layer serves as the loss layer that calculates the loss or error which represents a penalty for difference between the predicted and true labels. This layer normally is the final layer of a neural network. For predicting a single class of K mutually exclusive classes SoftMax loss is used. In various loss function appropriate it is a commonly used[19].

3.5 Model explanation

A simple CNN is a sequence of layers, and each layer transforms one volume of activations by a differentiable function from one to another. The three main layers explained before, convolutional layer, max pooling layer, and fully connected layer is used. These layers stack together to form the network.

Figure shows the proposed model architecture. First layer includes convolutional layer with activation function of ReLu. ReLu full name is Rectified Linear Unit, which applies the non-saturating activation function $f(x) = \max(0, x)$ [20]. ReLu help to remove the activation map negative value by setting them to zero[21]. First layer will get the input (pre-processed image) of size 28×28 pixel. In this layer, convolution filter size = 5×5 , padding = 0, stride = 1, and no. of filter = 32. A $32 \times 24 \times 24$ feature map is generated after the operation, 32 is the no. of feature maps, it also equals to the no. of filter used. According to the formula $((n+2p-f)/s)+1 = ((28+2*0-5)/1)+1=24$, the 24×24 is gotten. In every feature map the ReLu activation is complete.

After convolutional layer is max pooling layer, which also a second layer. The input is provided by the previous layer with the size of $32 \times 24 \times 24$. In this layer, pooling = 0, stride = 2, and size of pooling is 2×2 . A $32 \times 12 \times 12$ feature maps is produced after the max pooling executed. The number of feature maps remain unchanged after the operation due to the operation is independently executed in each feature map. The 12×12 pixel is got from the formula $((n+2p-f)/s)+1$. In addition, this layer has no activation function.

2nd convolutional layer is place in the third layer with ReLu activation function. In this layer the size of filter is 5×5 , padding = 0, stride = 1, and no. of filter = 32. An input ($32 \times 12 \times 12$) from previous layer is convoluted become $32 \times 8 \times 8$. Then ReLu activation is done in each feature map.

A max pooling layer is added again in the Layer-4. In this layer pooling size = 2×2 , padding = 0, and stride = 2. After max pooling operation, feature map size from $32 \times 8 \times 8$ (input) become $32 \times 4 \times 4$ (output).

Layer-5 is different from previous convolutional layer, which without ReLu activation function. A feature maps input with $32 \times 4 \times 4$ is got from previous layer. In this layer filter size = 4×4 , padding = 0, stride = 1, and no. of filter = 64. A $64 \times 1 \times 1$ feature maps is generated, after the operation. Layer-5 acts as a fully connected layer and generated 1-D vector (size 64) by being flattened.

Fully connected layer is located at Layer-6. In this layer an input of 1-D vector with the size 64 has been produce become a 1-D vector with the size 256. This layer has ReLu activation function.

Layer-7 will be the last layer, and it will also be a fully connected layer. The last layer would measure the class score, resulting in a vector of size 10, where each of the ten numbers corresponds to

the class score, similar to the 10 MNIST dataset categories. There's a SoftMax activation function for final outputs.

Follow this network architecture, the original image (input) with original pixel values layer by layer transforms to the final class scores using CNN. Some of the layers have parameters and some of them do not have. The parameters also been performed transformation by the convolutional or fully connected layer, not only for the activation in the input volume. In contrast, the ReLu or pooling layer will implement a fixed function. The class scores correspond to the training labels set for every image due to the parameters is trained with stochastic gradient descent algorithm in convolutional or fully connected layers[17].

4. Results and Discussion

This chapter provides the explanation of the result has been done of this project and discussion which included the combination of software and dataset used in this project. All the tool is installed to the computer. The MNIST dataset is import to the software phyton. The library needs for this project also been import to the software.

3.1 First model results

```
In [16]: 1 run_test_harness()
          > 98.692
          > 98.700
          > 98.583
          > 98.783
          > 98.517
```

Figure 1: The figure shows the good result from the model, every fold is achieved more than 98.00 % accuracy

To help in get an idea that the model evaluation is progressing, in the condition without shuffling by default the training and testing dataset is split into 5 consecutive folds.

The model performance is summarized by calculate the mean of the accuracy.

- Accuracy: mean=98.655 std=0.094, n=5

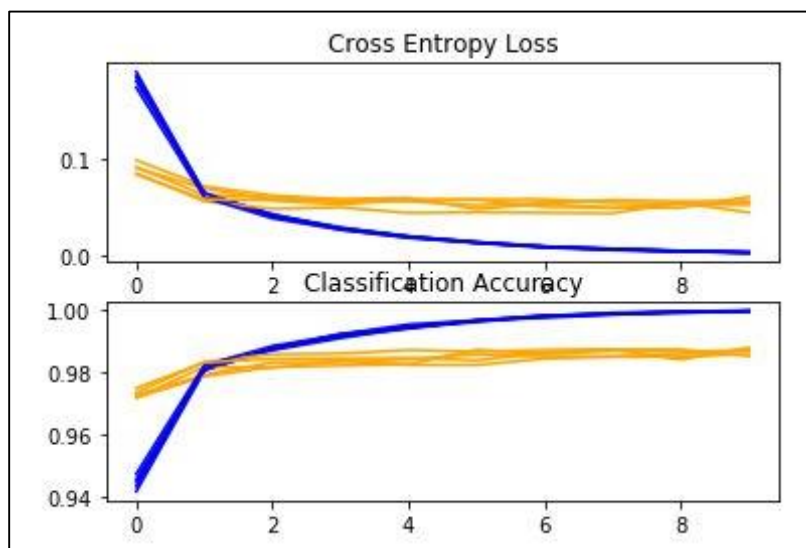


Figure 2: Figure show the diagnostic plot for loss and accuracy learning curves for the model during k-Fold Cross-Validation, blue line represents training dataset and orange represent testing dataset

3.2 Improvement of model

There are many approaches to increase the performance of the model. In this project, increase the model depth is used to improve the performance. Two 64-filters convolutional layer is added in the model, followed by another max pooling layer.

```

1 # entry point, run the test harness
2 run_test_harness()

> 98.892
> 99.150
> 98.992
> 98.800
> 98.975
    
```

Figure 3: The result shows the accuracy improvement on each fold

Accuracy: mean=98.962 std=0.116, n=5

Compare to the first model, the result shows some improvement of mean accuracy from 98.66 % to 98.962 %, with a small increase in the standard deviation from 0.094 to 0.104.

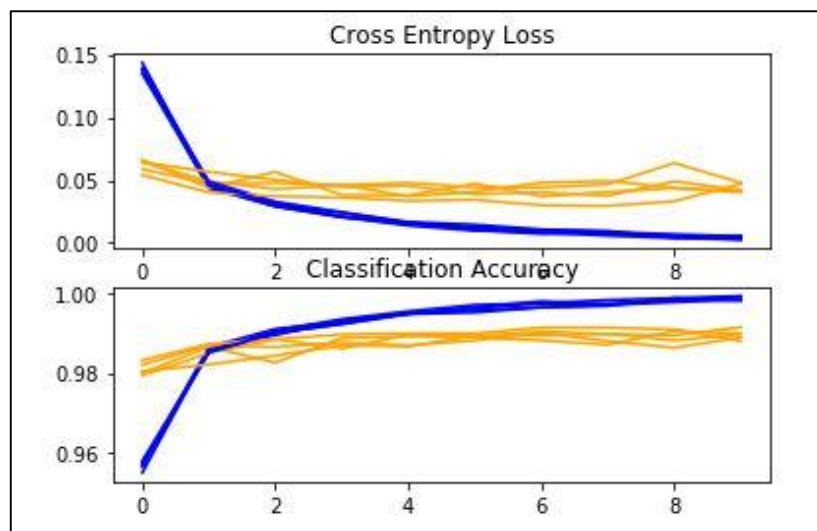


Figure 4: Diagnostic plot for loss and accuracy learning curves, blue lines represent training dataset and orange lines represents testing dataset

3.3 Final model

After comparing the first model and deeper model, the deeper model is chosen as the final model. The entire training dataset will be fit and save in a file for later use. The model then can be load and evaluate its performance on the Graphic User Interface [22].

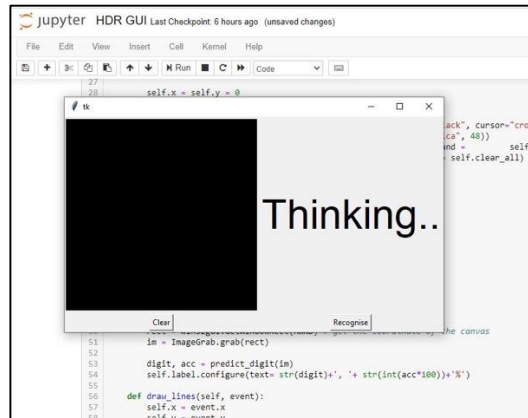


Figure 5: Figure show the GUI of this project. User can write digit on the black board

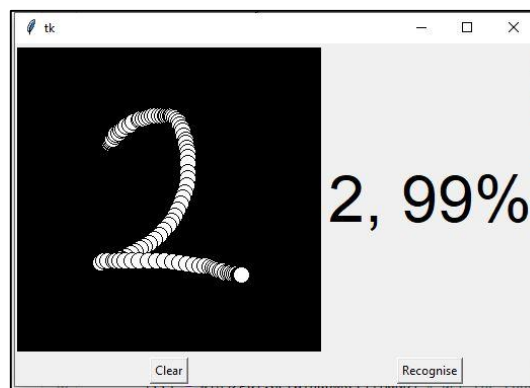


Figure 6: Right hand side show the prediction from the system and the percentage match for the label '2'

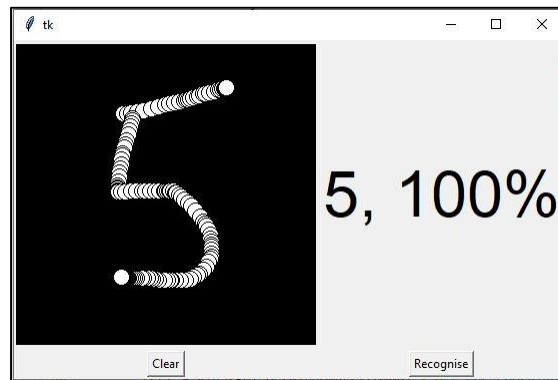


Figure 7: The system predicts the digit user provide is 5 and follow by 100 % match with label 5

3.4 Discussions

From the result observe, CNN is one of the good algorithms for handwritten digit recognition. For baseline model, it can achieve 98.00 % and above accuracy. There are many approaches to improve the model performance, one of them is tune the model depth which also the approach in this project used. Beside this tune pixel scaling and tune the learning rate also some of the ways can be used.

5. Conclusion

In this era of electronic data, Handwritten Digit Recognition system play an important role. It helps short the process to turn the data in digital form. Accuracy and time taken of recognition process is what the users care about. In this project the model builds with the objective to improve the accuracy and short the time taken to recognize. To data, phase 1 which is acquire and prepare data has been complete

by import MNIST dataset and reduce the pixel value of the data. The phase 2 building and compiling of the model is complete using CNN structure. Phase 3, training and evaluating the model is done. The model dataset is split into 5 consecutive folds for more accurate evaluation. An improvement approach is used to improve the model performance. In phase 4, a GUI is created for user-friendly in predict the output.

In this project, the result show CNN for Handwritten Digit Recognition provide a good performance. In baseline model, the model can provide a 98.00 % and above mean accuracy. From the result can be observed the model still have a lot of potential for develop. There still many ways for improve the model, example: tune pixel scaling, tune the learning rate, and so on.

Acknowledgement

This research was made possible by funding from research grant number H592 provided by Universiti Tun Hussein Onn Malaysia. The authors would also like to thank the Faculty of Engineering Technology, Universiti Tun Hussein Onn Malaysia for its support.

References

- [1] N. Rahal, M. Tounsi, T. M. Hamdani, and A. M. Alimi, "Handwritten Words and Digits Recognition using Deep Learning Based Bag of Features Framework," pp. 701–706, 2020, doi: 10.1109/icdar.2019.00117.
- [2] J. Qiao, G. Wang, W. Li, and M. Chen, "An adaptive deep Q-learning strategy for handwritten digit recognition," *Neural Networks*, vol. 107, pp. 61–71, 2018, doi: 10.1016/j.neunet.2018.02.010.
- [3] A. Holzinger, R. Geierhofer, and G. Searle, "Biometrical Signatures in Practice: A challenge for improving Human- Computer Interaction in Clinical Workflows," *Mensch und Comput.* 2006, 2015, doi: 10.1524/9783486841749.339.
- [4] S. W. Lee, *Advances in Handwriting Recognition (Series in Machine Perception & Artificial Intelligence)*. World Scientific Publishing Co Pte Ltd (1 July 1999), 1999.
- [5] Y. LeCun, C. Cortes, and C. J. C. Burges, "THE MNIST DATABASE of handwritten digits." .
- [6] R. O. Ph, *Introduction to Machine Learning, Second Edition (Adaptive Computation and Machine Learning)*. .
- [7] S. Ray, "A Quick Review of Machine Learning Algorithms," *Proc. Int. Conf. Mach. Learn. Big Data, Cloud Parallel Comput. Trends, Prespectives Prospect. Com. 2019*, pp. 35–39, 2019, doi: 10.1109/COMITCon.2019.8862451.
- [8] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
- [9] A. El-Sawy, E. Hazem, M. L.-I. conference on advanced, and undefined 2016, "CNN for handwritten arabic digits recognition based on LeNet-5," *Springer*.
- [10] I. R. B, A. Manfr, F. Vella, I. Infantino, and E. Lazkano, "Convolutional Neural Network vs Traditional Methods for Offline Recognition of Handwritten Digits," vol. 855, pp. 209–223, 2019, doi: 10.1007/978-3-319-99885-5.
- [11] N. Yu, P. Jiao, and Y. Zheng, "Handwritten digits recognition base on improved LeNet5," *Proc. 2015 27th Chinese Control Decis. Conf. CCDC 2015*, pp. 4871–4875, 2015, doi: 10.1109/CCDC.2015.7162796.
- [12] Prabhu, "Understanding of Convolutional Neural Network (CNN) — Deep Learning," *Medium*, 2018. [Online]. Available: <https://medium.com/@RaghavPrabhu/understanding-of->

- convolutional-neural-network-cnn-deep-learning-99760835f148. [Accessed: 04-Mar-2020].
- [13] E. Tuba, M. Tuba, and D. Simian, “Handwritten Digit Recognition by Support Vector Machine Optimized by Bat Algorithm,” *Conf. Comput. Graph. Vis. Comput. Vis.*, vol. 4617, pp. 369–376, 2016.
- [14] A. W. Chen, “Review Article Handwriting Recognition and Prediction Using Stochastic Logistic Regression,” vol. 05, pp. 5526–5527, 2018.
- [15] 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, doi: 10.17509/ijost.v3i1.10795.
- [16] C. Ma and H. Zhang, “Effective handwritten digit recognition based on multi-feature extraction and deep analysis,” *2015 12th Int. Conf. Fuzzy Syst. Knowl. Discov. FSKD 2015*, pp. 297–301, 2016, doi: 10.1109/FSKD.2015.7381957.
- [17] M. A. Hossain and M. M. Ali, “Recognition of Handwritten Digit using Convolutional Neural Network (CNN),” *Glob. J. Comput. Sci. Technol.*, vol. 19, no. 2, pp. 27–33, 2019, doi: 10.34257/gjcsdvol19is2pg27.
- [18] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” pp. 1–18, 2012.
- [19] N. Aloysius and M. Geetha, “A review on deep convolutional neural networks,” *Proc. 2017 IEEE Int. Conf. Commun. Signal Process. ICCSP 2017*, vol. 2018-Janua, pp. 588–592, 2018, doi: 10.1109/ICCSP.2017.8286426.
- [20] T. F. Gonzalez, “Handbook of approximation algorithms and metaheuristics,” *ImageNet Classif. with Deep Convolutional Neural Networks*, pp. 1–1432, 2007, doi: 10.1201/9781420010749.
- [21] V. V. Romanuke, “Flexible Solution of a 2-layer Perceptron Optimization by its Size and Training Set Smooth Distortion Ratio for Classifying Simple-Structured Objects,” *Res. Bull. Natl. Tech. Univ. Ukr. “Kyiv Politech. Institute,”* vol. 0, no. 6, pp. 59–73, 2017, doi: 10.20535/1810-0546.2017.6.110724.
- [22] “Deep Learning Project - Handwritten Digit Recognition using Python,” *Data Flair*, 2020. [Online]. Available: <https://data-flair.training/blogs/python-deep-learning-project-handwritten-digit-recognition/comment-page-1/>. [Accessed: 27-Dec-2020].