Digital Recognition by Deep Learning Techniques: A Proposed Digit Recognizer to Automate Cheque Deposition

Sook Chin Chiew[1], Xin Yuan Law[2], Ren Zhang Tan[3], XinYing Chew[4], Khai Wah Khaw[5*]
[1,2,3,4]School of Computer Sciences, UniversitiSains Malaysia, 11800 Penang, Malaysia
[5]School of Management, UniversitiSains Malaysia, 11800 Penang, Malaysia
[1]sookchin.chiew@student.usm.my,[2]lawxinyuan@student.usm.my,[3]renzhang@student.usm.my, [4]xinying@usm.my,
[5]khaiwah@usm.my

Abstract:
A cheque is a payment instrument that requires high-cost processing in banks because it involves significant manual works. The usage of cheque still exists as an important non-cash payment instrument,eventhoughBank Negara Malaysia has imposed a new processing fee of RM0.50 per cheque since 2015. In this paper, we propose a digit recognizer where manual input of payee's account number and cheque amount by the customer will be ceased to simplify the manual process at the cheque deposit machine. The handwritten account number and the cheque amount will be read by the digit recognizer automatically through the cheque image. The proposed automate process eases all the customers and reduces the recurring jobs done by bank staff which may cause input errors. The digit recognizer is developed usingConvolution Neural Network algorithm with preliminary accuracy of 99.65%.

Keywords: Cheque recognition, convolutional neural network, deep learning, digit recognizer

1. Introduction

In Malaysia, there are three basic types of the payment instrument used for a transaction, which included cash, cheque and electronic payment (e-payment). On the 2nd ofJanuary 2015, Bank Negara Malaysia (BNM) implemented a new pricing strategy on bank cheque, by imposing anew processing fee of RM0.50 per cheque [1]. The intention is to encourage customers to use e-payment and reduce paper-based payment instrument which involves manual process and high processing cost. According to Tan Sri Muhammad
Ibrahim, the BNM Governor, from 2011 to 2016, the cheque processing cost per unit increased from RM3 to RM4 and it is estimated to reach RM6 by 2020 [2]. Even with the rapid development of other types of payment instruments especially e-payment, cheque remains an important non-cash payment instrument[3]. Statistics obtained from BNM [4] in Fig. 1, show that the number of cleared cheques decreased significantly since 2013 due to the announcement of the new processing fee imposed onthecheque-based transaction[1].

However, 118.3 million cheques were cleared in 2017 [4] which shows that cheques are still widely used for the transaction. Processing bank cheques manually using methods such as manual reading, verify and input the cheque's information into the computer and so oncan be time-consuming thus leads to high labor cost and possible errors. Hence, an effective and accurate method is required so as to improve productivity. To this end, we propose a new solution that simplifies and reduces errors on cheque processing.

Ideally, there are two methodsto deposit a cheque, either through self-service cash deposit machine or usingthe bank counter. When using the cheque deposit machine, a customer requires to input manually the payee's account number and cheque amount into the machine as shown in Fig.2. The manual input process has been partially reduced by bank staff as compared to depositing a cheque through the counter. However, the manual input process is time-consuming for customers and there is a possibility of input error (human error) with this process. In the case of an error, more time is required to do the recurring job by bank staff as well as customers.

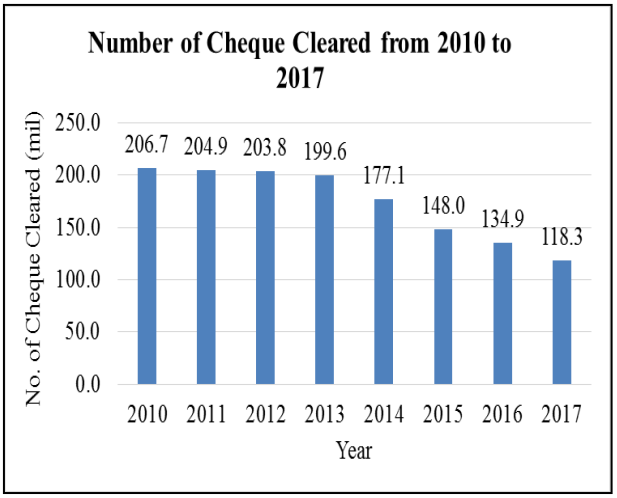

Fig. 1. Statistic of Cheque Cleared from 2010 to 2017 [4]

In this paper, we simplify the manual process done by customers, which is to input payee's bank account number and cheque amount manually to deposit cheques at the self-service machine. Customers are only required to insert cheques into the cheque deposit machine as shown in Fig.3,thereafter, the remaining process is automated.

Fig. 2.Process of cheque deposit at a self-service machine

Fig. 3.Proposed new process of cheque deposit at a self-service machine

We propose a new digit recognizer that can automatically retrieve the account number and courtesy amount that is written on cheque instead of being inputted manually by customers. Some researches in the past introduced different solutions to read the information on the cheque, which includes the courtesy amount (numeric), legal amount (textual), signature and particularly written language such as English, French or Korea for cheque processing automation or auxiliary verification [5]. In Malaysia, there is a research conducted to develop the Bank Cheque Recognition System by using the neural network, however, researchers were not satisfied with the performance obtained[6]. To the best of our knowledge, none of the research and implementation have been done on digital recognition in Malaysia, especially in improving performance in thebanking domain.

Our proposed solution aims to automate the cheque deposit process in Malaysia which will be beneficial to both bank staff and customers. Our proposed digit recognizer will incorporate with the cheque deposit machine. Customer isrequired to insert cheques into the machine and the machine will proceed to scan the image of cheque thenread the courtesy amount and bank account number based on the image that has been captured.
The account number and courtesy amount written on the cheque can be categorized into two types: printed form and handwriting form. Handwriting form will be the most challenging part for the computer to read as the handwritten style will vary for every person. In this paper, we focus on developing a digit recognizer that is can recognize handwriting digits on cheques to simplify the cheque deposit process. Recognition of handwriting digit is a complex problem compared to the common classification and regression problem. Hence, a deep learning algorithm will be used for the implementation of the digit recognizer. MNIST dataset [7],whichconsists of handwritten digits will be used to develop our digit recognizer.

The remainder of this paper is structured as follows: Section 2 reviews some literature in image processing using Artificial Neural Network and Convolutional Neural Network. Section 3 describes the implementation approach employed. Results and discussion are presented in Section 4. Section 5 concludes this paper while future works are presented in Section 6.

2. Literature Review

2.1 Digital Image Processing

Image processing is an approach used to extract the image content from several visual features such as color histogram, texture histogram, shape, object and relationship, and to learn the pattern of the image through image analysis. Image processing consists mainly of three parts, which are pre-processing, image analysis and post-processing. Pre-processing improves the performance of the image analysis. In this part, the image will be reconstructed and transformed into the format that eases and improves the image analysis. There are few tasks in this stage such as enhancing the visual qualities, compressing or resizing the image to ease the retrieving tasks; rotate the angle to make the object upright and so on. There are three types of image analysis: supervised or image classification, unsupervised or image clustering and reinforcement or finding their hidden associated rules. This paper will focus mainly on supervised learning (image classification). Post-processing is employed for the performance evaluation of the image model to guarantee its business value in implementing and making an explanatory report as well as information visualization. After constructing the classification model, the model will be tested and validated to ensure the efficiency of the model, so that the model is not overfitted or underfitted. The basic structure of image processing is shown in Fig.4.

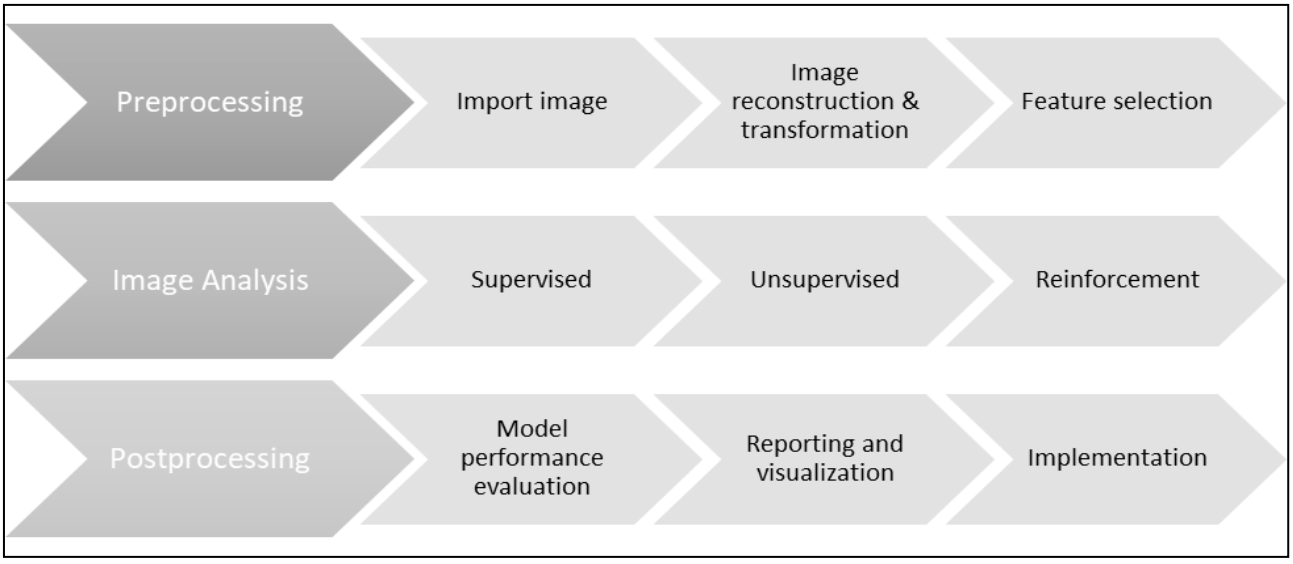

Fig. 4.The basic structure of image processing

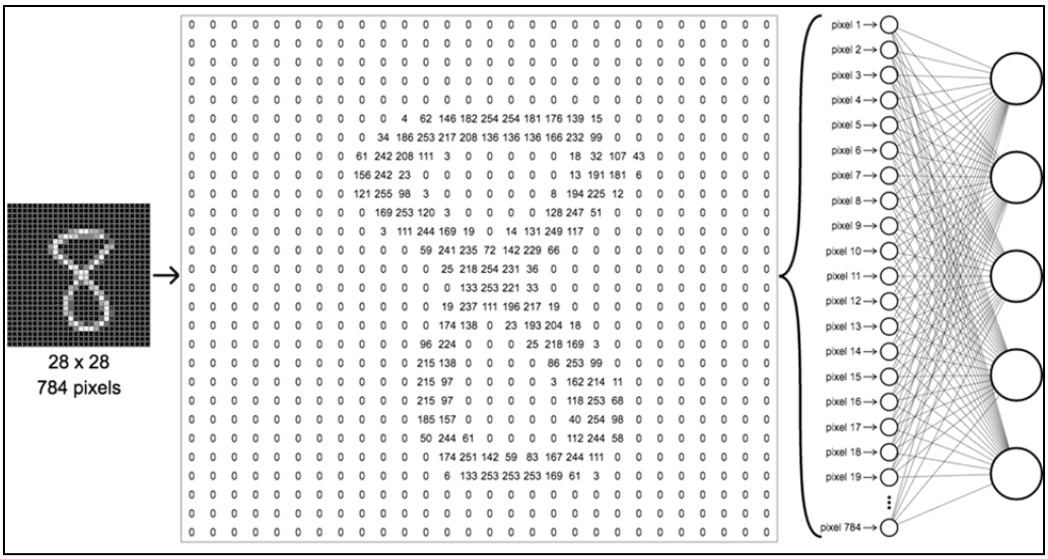

Fig. 5. Input into a neural network [8]

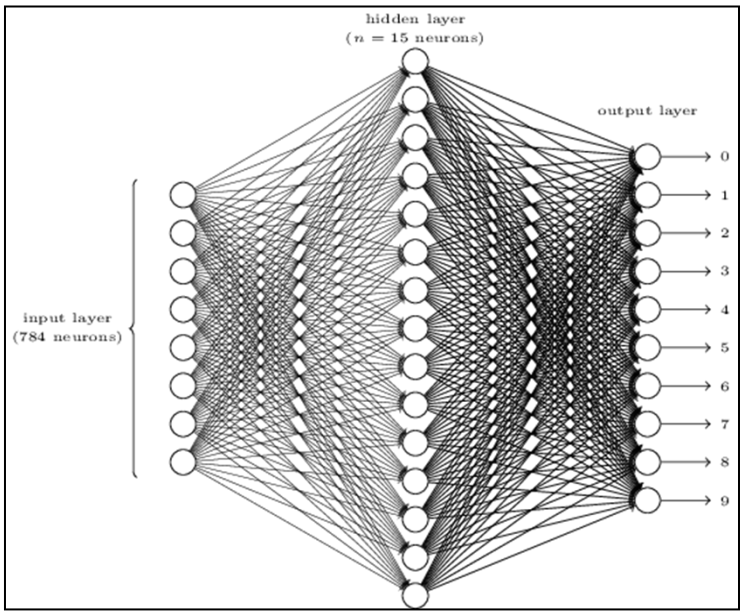

Fig. 6. The three-layers (input layer, hidden layer, and output layer) with its connection links [9]

2.2 Image Recognition

Image recognition is part of supervised learning in image analysis (that is, image classification). Image recognition is also known as visual interpretation, which identifies the digital images captured by computer vision technologies such as sensor, camera, scanner and so on. Image recognition is usually performed by deep learning. Deep learning is a subset of machine learning, which comprises algorithms for image recognition exposed by multiple layers of neural networks [10]. Deep learning transcends in recognizing objects in an image. The pattern of the image features will be learned, and the image will be classified through deep learning,Deep learning can be done by implementing artificial neural networks (ANN) to extract visual features from the image [11]. One of the fundamental visual features used for handwritten digits recognition is to assign greyscale for each pixel on an image. Thereafter, all the greyscale values are stored into an array for future computations in the image classification. Each artificial neuron in neural networks will represent a greyscale value for each pixel on an image as depictedin Fig.5.

2.3 Artificial Neural Networks (ANN)

ANN is the simplest neural networks in deep learning. Basically, the collection of artificial neurons is organized into three main layers, which are the input layer, the hidden layer, and the output layer[12].Each neuron undergoes computation with mathematical formulae such as weighted sum, sigmoid or tanh, from layer to layer. Connection acts as a bridge to join two layers, where each connection link carries a certain weight, ω and bias, ε for the input neuron, x_i from the current layer to compute the value for the output neuron, y_j at the next layer.Normally, the computation will generate the weighted sum function [13] as written below:

$$y_j = \sum_i (\omega_i \cdot x_i) + \varepsilon_j$$

Instead of using the weighted sum function, there are many types of activation function such as ReLU[14]. The structure of basic three-layer neural networksconsisting of one input layer, one hidden layer, and one output layer is as shown in Fig.6.

As all the neurons are computed by the function, ANN has the possibility of overfitting. WithANN, it is hard to understand the learned layer due to the dense connection that fully connects all neurons. Therefore, ANN should be improved by using different activation function, weight optimization, regularization, or noise reduction. Besides, more layers can be added to the filter and aggregate the image information carried by each image pixel.

2.4 Convolutional Neural Networks (CNN)

Generally, a deep neural network is made up ofmultiple hidden layers. The deeper the depth of the networks, the more accurate the performance of the recognition, meanwhile it takes longer time to train the model as well[15]. One of the deep learning algorithms that are suitable and commonly used for handwritten digit recognition is CNN [16]. CNN is spatially or fully connected between two consecutive layers, where each output neuron computes from a subset of the input neurons based on proximity. While passing through multiple layers, it sacrifices some generalizability for attaining a much more feasible solution[17]. In this case, the added computational load makes our network less sophisticated and thus minimizescomputational cost.

CNN filters the raw pixel data from the input neuron via five types of layers (Fig. 7):

1. Convolutional layers filter the input neurons, yield a single output neuron for each subset, and then apply a ReLU activation function to the output neuron to acquaint nonlinearities.
2. Pooling (down-sampling) layers downsample the input neurons from the convolutional layers to reduce the dimensionality of the features and decrease its processing time. A commonly used pooling algorithm is max pooling, which extracts the maximum value from each subset of the input neurons and discards all other values [19].

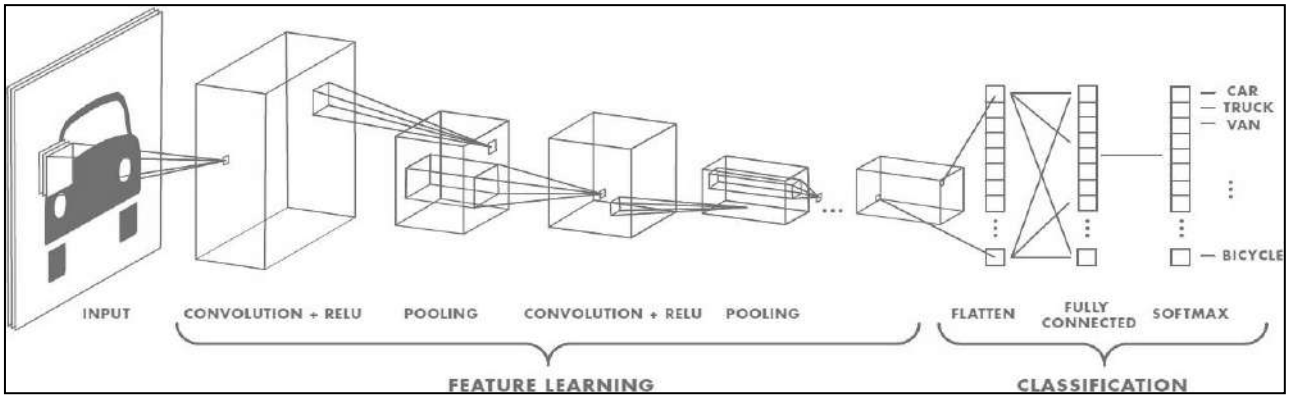

Fig. 7. Layers in a typical CNN for image classification [18]

3. Dropout layers randomly drop out neurons from the network. The network becomes less sensitive to the specific weights of neurons so that the network achieves better generalization and avoid overfitting.
4. Flatten layers convert all two-dimensional arrays into a single long continuous linear vector.
5. Dense (fully connected) layers classify the features. In the final dense layer, a single neuron is assigned for each target class in the CNN model by using a softmax activation function. The softmax values of an image are the relative measurements of how likely the image falls into each target class [20].

An artificial neuron is calculated using the weighted sum function to decide whether it should activate or not. For convolutional neural networks, there are few activation functions that can be considered throughout the layers:

1. ReLU function [21]

$$f(x) = \begin{cases} x & ,if\ x > 0 \\ 0 & ,if\ x \leq 0 \end{cases} \qquad (2)$$

ReLu is ranging in [0, ∞). It is half rectified and therefore the network is smaller, therefore it does not train the negative values appropriately.

2. Softmax function [21]

$$\sigma(x)_i = \frac{e^{x_i}}{\sum_i e^{x_i}}$$

Softmax calculates the probabilities of each target class over all possible target classes. It results in a probability range of [0,1] and the sum of all the probabilities will be equal to one.

A simple comparison between artificial neural networks and convolutional neuralnetworks is summarized in Fig.8.

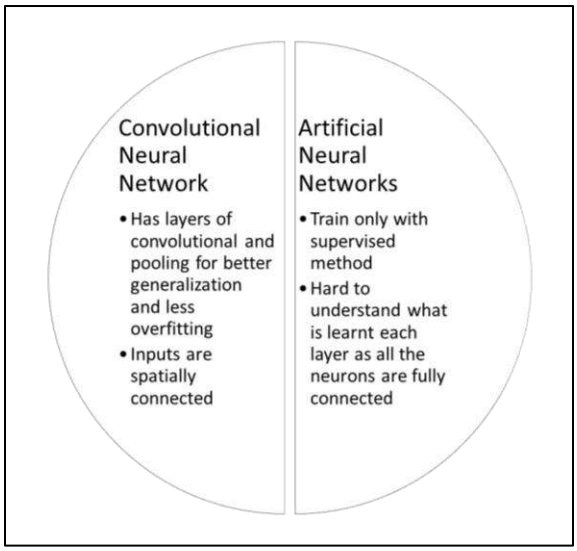

Fig. 8.The comparison of convolutional neural networks and artificial neural networks

By taking into consideration the factor of overfitting and complexity of the algorithm, we selected CNNasour algorithm to be used in this paper.

- Overfitting
  CNN works well in generalizationwithless overfitting as compared to ANN which has a higher probability of overfitting.

- Complexity
  CNN has lower complexity which reduces the resources required for computation. In addition, the lower complexity makes it apreferred algorithm for understanding the learned layer.

3. Methodology

Python is an open source, a high-level programming language for general-purpose programming. Due to its user-friendliness, simplicity and fast prototyping, it has become the most popular programming language among data scientists. It has powerful statistical and numerical packages e.g. numpy and pandas to read andmanipulate data accuratelyandefficiently. It also has a handy machine learning package calledscikit-learn. In this paper, we are going to build a CNN model using the aforementioned packages. Python has Tensorflow and Keraswhichcan be used for the development of an efficient deep learning model. The extensive libraries show that python has mature support for data science, hence we selected Python as our implementation language to perform data analytics and machine learning in this paper.

Our proposed system consists of two processes. First, we need to recognize the regions that contain digits in a cheque. We are interested in extracting digits from 3 fields (i.e. date, amount and account number). This step involves recognizing the interested fields in a cheque and extracting the handwritten digits into individual images. Second, we need to train a model to recognize the images of handwritten digits and classify them accordingly. In this section, we will start with the process of building and training a CNN model for the digit recognizer, then proceedto digits extractions from the cheque.
MNIST dataset has a good collection of handwritten digits. The train set contains 60,000 rows of data whereas the test set contains 10,000 rows of data. The binary data is available at Yann's home page [7]. In this paper, we will use MNIST dataset to build out digit recognizer as a proof of concept. To demonstrate the real-life situation where the input is a real image, we have extracted 70,000 PNG images [22] and converted them into the 28*28 matrix in numpy. Fig.9 shows the shape of a sample image, its real representation, and matrix visualization.

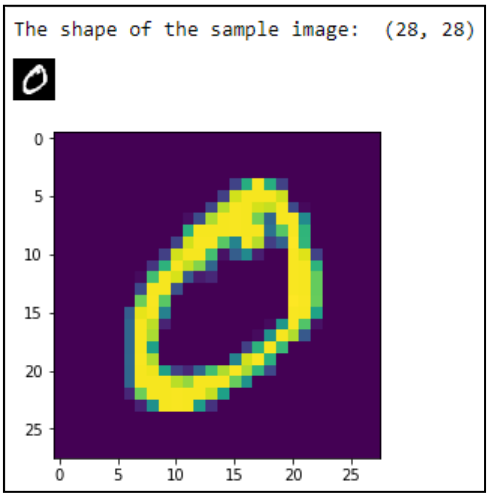

Fig. 9.A random example from MNIST

MNIST has a balanced distribution of class labels. The difference ineach class label is not significant. It is a good dataset to study pattern recognition method on real-world data.

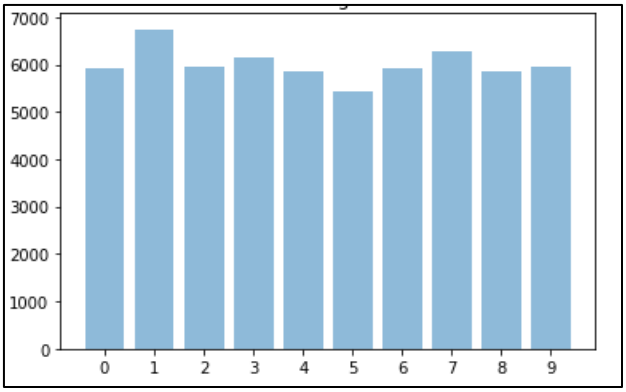

Fig. 10.The bar chart shows the distribution of the class label (0 to 9)

In this paper, Keras and Tensorflow are used to build our CNN model. Keras is an open source, high-level neural network library. It provides a set of APIs to run different neural network backbones such as Tensorflow and Theano. Tensorflow is an open source library for numeric computation and dataflow programming developed by the Google Brain team. It uses the dataflow graph to represent

complex mathematical operations on multidimensional data arrays. Tensorflow is particularly useful in designing deep neural network models[23]. While Tensorflow is harder to interpret, Keras provides a user-friendly layer on top of Tensoflow. It optimizes the modeling process and reduces the complexity associated with experiments.
The sequential API in Keras provides a method to model the neural network layer by layer. It is a linear stack of layers. Depending on the design, the layers will vary for each problem. In this paper, our CNN starts with an input layer to accept 28*28*1 array, followed by2 convolutional layers with 32 bits filter and 6*6 kernel size. This means that these layers will scan through the 6*6image and the output would be a 32 convolved feature set. A typical activation function selected for the convolutional layer is ReLu.

The fourth layer is a pooling layer for downloading sampling purpose. With the pool size of 2*2, it will look at the neighboring pixels, pick the maximal value and eliminate the minimal value. Pooling layer helps to reduce the feature dimensionality, which can improve the computational efficiency.

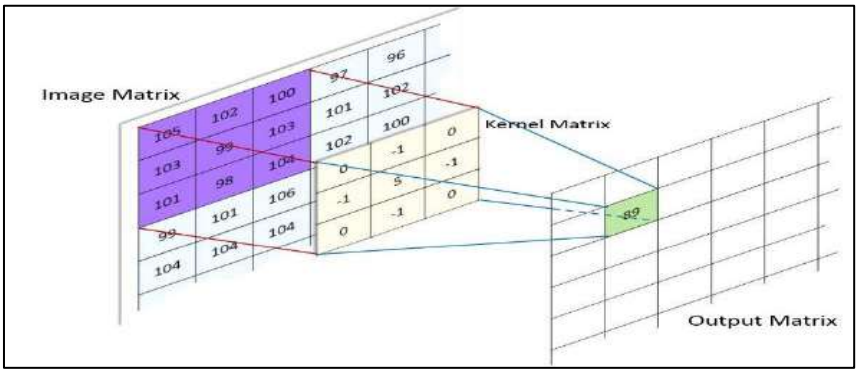

Fig. 11.Conv layer extracts the output matrix from the input matrix[24]

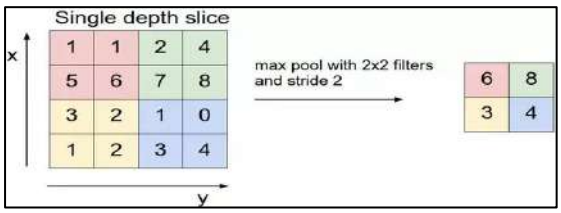

Fig. 12.Sampling the neighboring pixel and select the maximal value[25]

In the fifth layer, drop out is a technique to prevent over-fitting in the neural network. By randomly ignoring some neurons during the training process, the accuracy will be decreased on the training set. It trades the training performance for better generalization in test set or future data. This CNN model willbe a generalized model to predict unseen data, not only fit into the MNIST dataset.

Once the features are extracted and learned, flatten layer will turn this 3d matrix into a 1d matrix and prepare for classification. The output layer is a dense layer with a softmax activation function. The output is an array with 10 binary variables, indicating the predicted label (0 to 9) of the input data. With such a model, the accuracy on the test set was 98% in 10 epochs.

To improve the model performance, data augmentation is applied to this model. The key idea is to artificially transform MNIST data and make the existing dataset even larger. It will also help to avoid the issue of over-fitting. The chosen settings are randomly zoomed in the image by 15%, shifted horizontally by 10% and shifted vertically by 10%. Moreover, few more layers are added to our model. Deeper convolutional and pooling layers will look at a smaller region and extract even more features. Fig. 14 shows the architecture of our final model.By utilizing GPU processing on NVIDIA CUDA, the time spent of each epoch is shorter from >= 150 seconds on Intel i7 7700 to 10 seconds on GTX 1060 3GB.The final accuracy is 99% on the train set, 99.17% on the validation set and 99.65% on the test set.Fig. 15 shows the confusion matrix on the test set. This trained model will be stored into .h5 format for later usage.

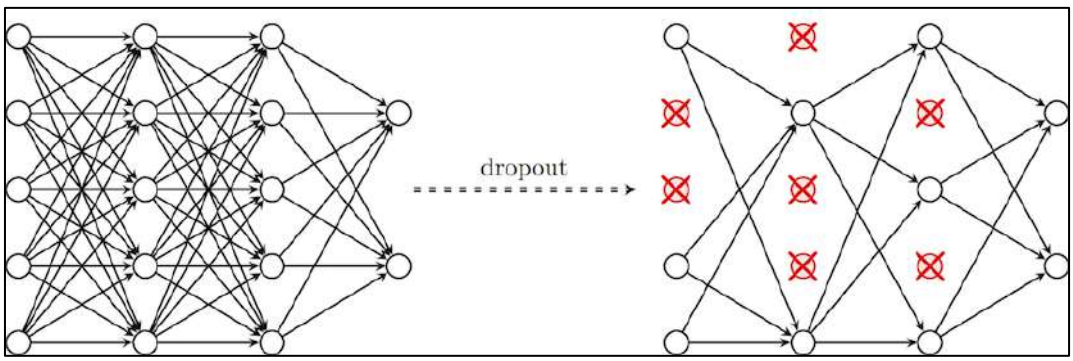

Fig. 13.Randomly ignore some nodes by dropout layer[26]

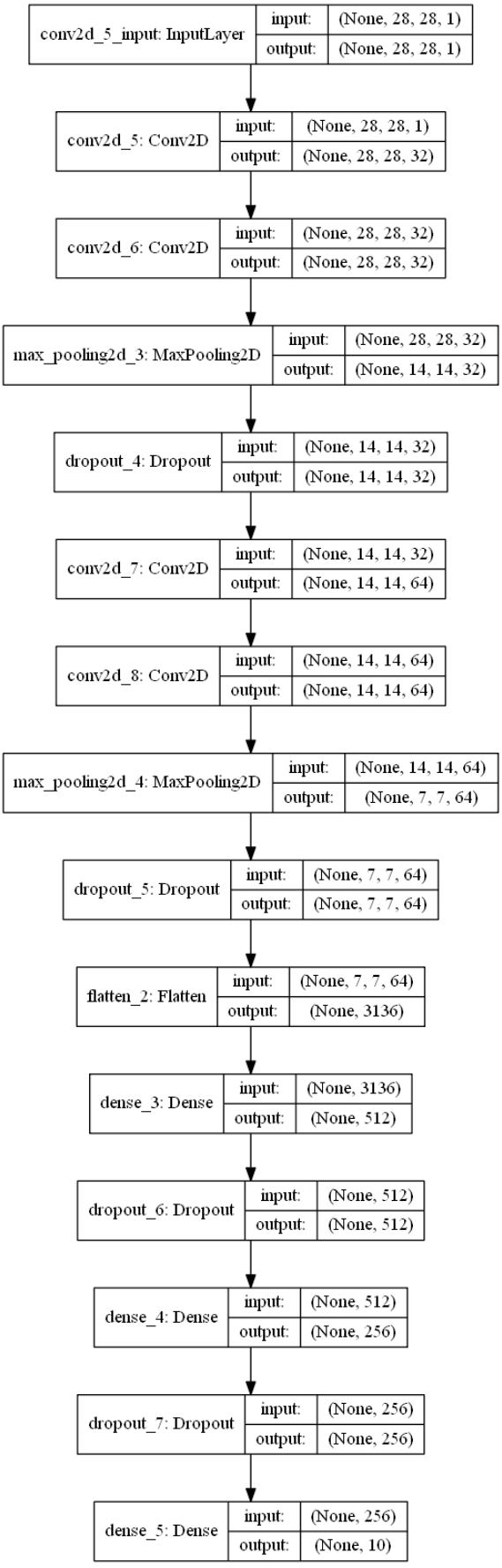

Fig. 14.The architecture of our CNN model

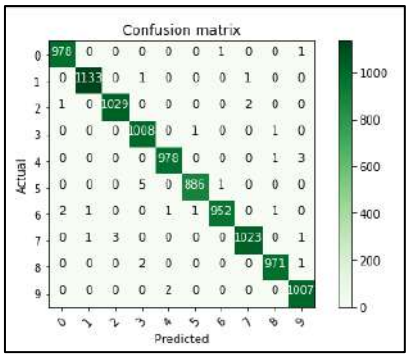

Fig. 15.Confusion matrix on the test set

Once the model is ready, we can proceed to digits extraction from a cheque. To mimic a real-life situation, a cheque template is created for this paper. We manually write the amount, date and account number. After that, both sides of the cheque will be captured by using a phone's camera. In the banking industry, they use a dedicated scanner to scan a cheque into the system. This provides aclearerimage than a camera photo without affecting by the light condition.

Fig. 16.Cheque template, front and back side

First, we start with the front side of the cheque. We are interested in two particular parts: date and amount field. Both have the same background (i.e. light gray color). At this point, we will do a step by step cropping to retrieve the interested fields. First of all, we need to extract the main body of this cheque with yellow color and remove irrelevant contents. This can be done by using the "find contour" in OpenCV. We can filter out the interested area by using a mask. The mask here is in yellow color. Once we recognize the main part of the cheque, then we can extract the largest and second largest contours in that image by using a light gray mask. The largest contour is the amount field, and the second largest contour is the date field. Fig. 17 shows the trimmed cheque, the amount field, and date field.

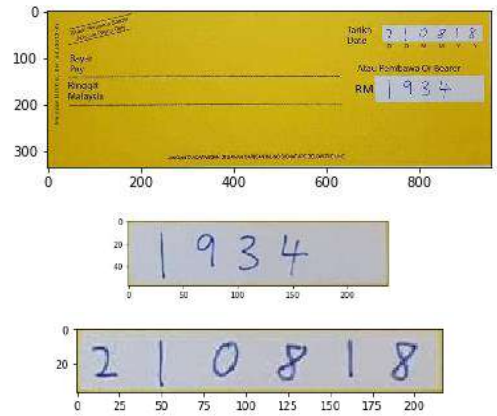

Fig. 17.Trimmed cheque front side, amount field, and date field

For the back side, we are interested in the account number. Same as the front side, we trim the cheque by finding the largest yellow contours, then we select the second largest contour to determine the account number field. The reason wechoosethe second largest is that the largest light gray region is the reserved areaforthe appropriate bank official to write something.

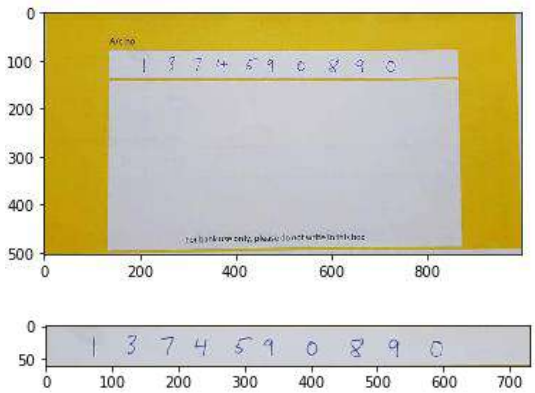

Fig. 18.Trimmed cheque back side and account number field

Colour does not matter in our digit recognizer, so we will convert the 3 interested fields into a grayscale image.

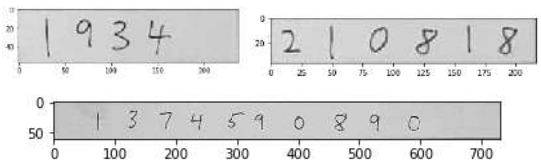

Fig. 19.Amount, date and account number in grayscale

Our CNN model is trained using the MNIST dataset i.e. the digit in white and background in black color. Now we convert our images into that pattern.

We are going to extract each individual digit from the images. By using find contour, we can know how many digits are present in an image. It will split the images based on the boundary of white color. Furthermore, we filter out the image that is smaller than 3x3 because it is likely to be noise and not a digit. The digits will be sortedusing x-axis to get the original sequence. For each digit image, black color padding will be added to centralize the digit. Moreover, all images will be resized into 28x28 to match the size of the MNIST images. Fig.21 shows the first character extracted from each field.

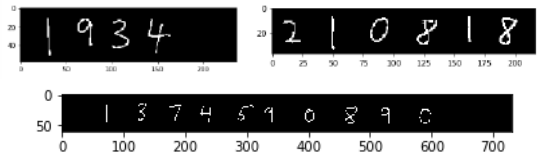

Fig. 20.Amount, date and account number in black and white

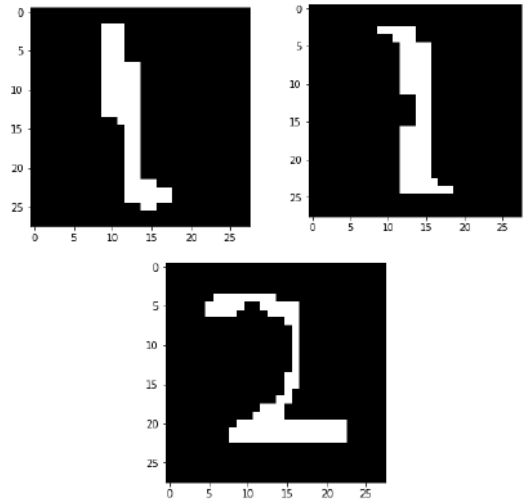

Fig. 21.First digit for amount, date and account number.

Once all individual digits are extracted into separate images, we thenconvert it into a numpy array and put into our pre-trained CNN model for classification. Below is the predicted value of each field versus the actual value.

	Actual	Predicted
Amount	1**9**34	1**7**34
Date	210818	210818
Account No	13745**9**0890	13745**7**0890

From the result, we can see that this model has an issue in predicting 9. It misclassified two of the '9s' into '7s'. To further improve the result, we can put more training data for '9' and '7' to re-train the model. The overall performance in this experiment is 18/20 = 0.9. Fig.22 shows the 2 misclassified images for '9'.

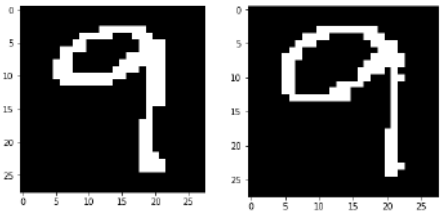

Fig. 22. Two misclassified images, actual is 9 but predicted is 7

4. Result and Discussion

We used the open source tool, Python which contains useful libraries and CNN algorithm to develop our digit recognizer model. Using OpenCV, image pre-processing was applied to extract the series of digits from a cheque image. Overall, the model performed very well with an accuracy of 99.65% on the test dataset and 90% for our experiment were on sample cheques.By using GPU, the model performedeven betteron computation as the training time was approximately 100 seconds and less than 3 seconds for model evaluation.

In this paper, our objective wastosimplify the manual process of cheque deposition, by developingadigitrecognizerwhichis capable of reading handwriting digit.We have shown that it is feasible to replace the process of manual input payee's bank account number and cheque amount during cheque deposition. With referenceto the obtained result, the accuracy of our model shows the efficiency of the proposed digit recognizer. However, the reliability of our modelcan beimprovedusingreal-world digit

Fig. 23.Proposed new process of cheque deposit with build-in digit recognizer at self-service cheque deposit machine

image from the cheque. Using only MNIST dataset to train our model is not enough for our use case. Fig. 23 illustrates the proposed process of cheque deposit with build-in digit recognizer at self-service cheque deposit machine.

5. Conclusion

In this paper, a system is developed to extract the digits from cheque and which are classified using the CNN model. The accuracy of our CNN model is 99.65% while testing on MNIST test dataset, and 90% while testing on a sample cheque. Given a cheque image, this application is able to recognize fields of interest, as well as extract the digits accordingly and classify the value of the series of digits. This application can help the banking industry improve its cheque deposit process. However, no mistake is allowed during this process. A single mistake from the system can cost the bank a lot financially. If the wrong amount of money is being transferred or money is transferred tothewrong account number, the bank may lose its trust and reputation by clients. In practice, accuracy should be more than 99.999%. This can be improved by increasing the digit sequenceof the training datasetissued to CNN model.

REFERENCES

[1] Central Bank of Malaysia. (2014, March 19). New Pricing for Cheques will now take effect on 2 January 2015. Retrieved 23 April 2018, from http://www.bnm.gov.my/index.php?ch=en_press&pg=en_press&ac=557&lang=en

[2] J. Chin, (2017, August 12). Cheques to cost more as Bank Negara boosts e-payments - Business News | The Star Online. Retrieved 23 April 2018, from https://www.thestar.com.my/business/business-news/2017/12/08/cheques-to-cost-more-as-bank-negara-boosts-epayments/

[3] Central Bank of Malaysia. (n.d.-b). Payment Systems in Malaysia | Bank Negara Malaysia | Central Bank of Malaysia. Retrieved 23 April 2018, from http://www.bnm.gov.my/index.php?ch=ps&pg=ps_mps_type&ac=177&lang=en

[4] Central Bank of Malaysia. (n.d.-a). Monthly Highlights and Statistics February 2018 | Bank Negara Malaysia | Central Bank of Malaysia. Retrieved 23 April 2018, from http://www.bnm.gov.my/index.php?ch=en_publication&pg=en_msb&ac=255&lang=en&uc=2

[5] R. Palacios, and A. Gupta, "A system for processing handwritten bank checks automatically,"*Image and Vision Computing*, 26(10), pp. 1297–1313, 2008.

[6] A. Wahap, M. Khalid, A. Ahmad, andR. Yusof, "A Neural Network Based Bank Cheque Recognition system for Malaysian Cheques", 2002.https://www.researchgate.net/publication/229001364_A_Neural_Network_Based_Bank_Cheque_Recognition_system_for_Malaysian_Cheques?enrichId=rgreq-d167e08a86ea4f07e1e8f19835de9e92-XXX&enrichSource=Y292ZXJQYWdlOzIyOTAwMTM2NDtBUzoxMDUwMjU3NjIyMzQzNzJAMTQwMjA1MTUwMTg1Nw%3D%3D&el=1_x_3&_esc=publicationCoverPdf

[7] Y. LeCun, C. Cortes, and C. Burges, (n.d.). MNIST handwritten digit database. Retrieved 25 April 2018, from http://yann.lecun.com/exdb/mnist/

[8] G. Kogan, (2018). *ml4a.github.io: machine learning for artists*. HTML, ml4a. Retrieved 24 April 2018, from https://github.com/ml4a/ml4a.github.io (Original work published 2016)
[9] M.A. Nielsen, (2015). Neural Networks and Deep Learning. Retrieved 24 April 2018, from http://neuralnetworksanddeeplearning.com
[10] R. Parloff, (2016, September 28). Why Deep Learning Is Suddenly Changing Your Life. Retrieved 24 April 2018, from http://fortune.com/ai-artificial-intelligence-deep-machine-learning/
[11] A. Abdelfattah, (2017, July 27). Image Classification using Deep Neural Networks — A beginner friendly approach using TensorFlow. Retrieved 24 April 2018, from https://medium.com/@tifa2up/image-classification-using-deep-neural-networks-a-beginner-friendly-approach-using-tensorflow-94b0a090ccd4
[12] S. Miller, (2015, October 8). Mind: How to Build a Neural Network (Part One). Retrieved 8 June 2018, from https://stevenmiller888.github.io/mind-how-to-build-a-neural-network/
[13] S.V. Avinash, (2017, March 30). Understanding Activation Functions in Neural Networks. Retrieved 24 April 2018, from https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0
[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning". *Nature*, 521(7553), pp. 436–444, 2015.
[15] W. Yang, Q. Liu, S. Wang, Z. Cui, X. Chen, L. Chen, and N. Zhang, "Down image recognition based on deep convolutional neural network,"*Information Processing in Agriculture*, 5(2), pp. 246-252, 2018.
[16] M.D., McDonnell, M.D. Tissera, T. Vladusich, A. van Schaik, and J. Tapson, "Fast, simple and accurate handwritten digit classification by training shallow neural network classifiers with the 'Extreme Learning Machine' algorithm,"*PLOS ONE*, 10(8), e0134254, 2015.
[17] T. Keenan, (2017, May 1). How Image Recognition Works. Retrieved 24 April 2018, from https://www.upwork.com/hiring/data/how-image-recognition-works/
[18] The MathWorks. (n.d.). Convolutional Neural Network. Retrieved 25 April 2018, from https://uk.mathworks.com/solutions/deep-learning/convolutional-neural-network.html
[19] The Comprehensive R Archive Network. (n.d.). TensorFlow Layers. Retrieved 24 April 2018, from https://cran.r-project.org/web/packages/tfestimators/vignettes/tensorflow_layers.html
[20] TensorFlow. (2018, March 29). A Guide to TF Layers: Building a Convolutional Neural Network. Retrieved 24 April 2018, from https://www.tensorflow.org/tutorials/layers
[21] Kulbear. (2018). *deep-learning-nano-foundation: Udacity's Deep Learning Nano Foundation program*. Jupyter Notebook. Retrieved 24 April 2018, from https://github.com/Kulbear/deep-learning-nano-foundation (Original work published 2017)
[22] M. Ott, (2018). *mnist_png: MNIST converted to PNG format*. Python. Retrieved 10 March 2018, from https://github.com/myleott/mnist_png (Original work published 2015)
[23] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, … X. Zheng, "TensorFlow: A system for large-scale machine learning," In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*, 2016.
[24] Machine Learning Guru. (n.d.). Image Convolution. Retrieved 25 April 2018, from http://machinelearninguru.com/computer_vision/basics/convolution/image_convolution_1.html
[25] D. Britz, (2015, November 7). Understanding Convolutional Neural Networks for NLP. Retrieved 25 April 2018, from http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/
[26] P. Veličković, (2017, March 20). Deep learning for complete beginners: convolutional neural networks with keras. Retrieved 25 April 2018, fromhttp://cambridgespark.com/content/tutorials/convolutional-neural-networks-with-keras/index.html