Date of publication SEP-30, 2023, date of current version APR-15, 2023.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.22.3.3232

Classification of Letter Images from Scanned Invoices using CNN

DESIREE JUBY VINCENT, HARI V. S.
Department of Electronics, College of Engineering Chengannur (Affiliated to APJ Abdul Kalam Technological University), Chengannur, 689121, Kerala, India

Corresponding author: Desiree Juby Vincent (e-mail: jubyin@gmail.com).

This work was supported by Research Program supported by CERD (Centre for Engineering Research and Development)funding of APJ Abdul Kalam Technological University, and FIST (Fund for Improvement of Science and Technology Infrastructure in Universities and Higher Educational Institutions), College of Engineering Chengannur, Kerala.

ABSTRACT Data analytics helps companies to analyze customer trends, make better business decisions and optimize their performances. Scanned document analysis is an important step in data analytics. Automatically extracting information from a scanned receipt has potential applications in industries. Both printed and handwritten letters are present in a receipt. Often these receipt documents are of low resolution due to paper damage and poor scanning quality. So, correctly recognizing each letter is a challenge. This work focuses on building an improved Convolutional Neural Network (CNN) model with regularization technique for classifying all English characters (both uppercase and lowercase) and numbers from 0 to 9. The training data contains about 60000 images of letters (English alphabets and numbers).This training data consists of letter images from windows true type (.ttf) files and from different scanned receipts. We developed different CNN models for this 62 class classification problem, with different regularization and dropout techniques. Hyperparameters of Convolutional Neural Network are adjusted to obtain the optimum accuracy. Different optimization methods are considered to obtain better accuracy. Performance of each CNN model is analyzed in terms of accuracy, precision value, recall value, F1 score and confusion matrix to find out the best model. Prediction error of the model is calculated for Gaussian noise and impulse noise at different noise levels.

KEYWORDS CNN (Convolutional Neural Network); L 2 Regularization; Early stopping; OCR (optical character recognition)

I. INTRODUCTION

DESPITE the increase in digital transactions, there is a large share of manual transactions and associated paper documentation in developing countries. It is required in many cases to extract the information in paper documents such as scanned receipts. This fact is especially true of old documents and manuscripts which originated before the era of digitization. Such extracted data are crucial in office automation, accounting, financial, taxation and supply chain management. The optical character extraction and recognition are challenging tasks in computer vision especially when the documents are scanned or photographed. Data from scanned documents [1] are useful in making business decisions, by understanding customer behavior and needs, to offer better products and services to them. The present work focuses on the development of CNN models for recognition of English letters and numbers from scanned receipts.

Optical character recognition (OCR) is a system for converting input letter images into digital format that a machine can understand. It can convert both typewritten documents and handwritten documents to digital format. OCR system is used for the extraction of features and then classification. Optical character recognition is a science that enables to translate various types of documents or images into analysable, editable and searchable data. A summary of the research conducted on character recognition of handwritten documents was given in [2]. A historical review of OCR research and development was described in [3]. A document-specific character recognition (OCR) system using an appropriate clustering algorithm was presented in [4]. A domain-specific knowledge based OCR system was given in [5]. An OCR system for 10 widely used font styles of upper and lower case letters, with reduced computational complexity, was presented in [6]. But this method suffers from a low classification accuracy of only 80%. Statistical methods of information retrieval for the classification of German business letters were given in [7]. A tree classifier, followed by template-matching approach, for Bangla character classification was detailed in [8].

Research was conducted on handwritten digit recognition [9] that was a challenging task due to the different writing styles. It has been observed that the ensemble method has the least error rate. An improved CNN with enhanced digit recognition capability was presented in [10]. In the latter work, a large data set of handwritten digits was used for training the network to obtain the spatial characteristics of handwritten digits. An offline recognition system for handwritten digits, based on LeNet-5 in the convolutional neural network, is used for improved feature extraction [11]. Pre-processing of samples and the use of ensemble model resulted in improved accuracy [12]. Hidden Markov model (HMM) based recognition of handwriting was explained in [13]. These HMMs are concatenated to form letter models, which are then embedded in a stochastic language model. A classification, which combines dynamic time warping (DTW) and support vector machines (SVMs) by establishing a new SVM kernel, is used for online handwriting recognition [14]. Deep learning methods have the advantage that they can automatically do both the feature extraction and classification and can achieve high accuracy beyond the human level performance. They are widely used in one dimensional and two dimensional pattern recognition applications [15-18]. OCR system for printed text documents in Kannada, a South Indian language using 2-class classifiers is based on the Support Vector Machine (SVM) method [19]. A Convolutional Neural Network (CNN) based Optical Character Recognition system (OCR) which accurately digitizes Ancient Sanskrit manuscripts (Devanagari Script) was explained in [20]. A system for classification of source printer from scanned images of printed documents using all the printed letters simultaneously was detailed in [21].

The main challenge in letter recognition from scanned receipt is the presence of noise which degrades the periphery of letters. Besides, receipts are printed with different font shapes and styles. However, most receipts are generated with windows true type fonts whose statistics are known. The letter images, generated with these fonts, are used to train character recognition systems. Deep learning methods, which can automatically do both the feature extraction and classification, are widely employed for optical character recognition (OCR).

II. THEORY

Letter recognition is the capacity of a computer to interpret different characters and numbers. Printed characters in receipts may be of different font styles. Manually written letters in receipts are not of a similar size, thickness and position. The composition styles of various individuals also affect the style of letters. Accuracy of OCR system increases, if all these types of various letter images are included in a training set. In our work, the training data contains about 60,000 images of letters (English alphabets and numbers) obtained from windows true type (.ttf) image files and from various scanned receipt images.

Deep learning based convolutional neural networks (CNN) [22] are observed to yield good results in image recognition applications. CNN is appropriate to represent the image structure, because of its local connectivity strategy, and the weight sharing property. CNN is widely used in character recognition, document analysis and receipt data recognition [23-26]. But most of these models have low accuracy for external dataset, due to lack of dedicated training sets of receipt data images. CNN model for letter image classification and recognition is made to perform better by tuning the various hyperparameters, and selecting the proper optimization algorithms [27]. By employing large datasets and regularization methods, better model performance is achieved.

A. CONVOLUTION NETWORK

In case of images, which has grid like topology, CNN can learn the features in an efficient way with less number of neurons. The spatial relationship between pixels in an image helps the filters in CNN to obtain local connectivity to a neuron with these pixels. CNN can effectively learn complex patterns. CNN architecture consists of a series of convolution layers, which performs convolution operation, followed by activation functions and maxpooling operations. Dense layers that are fully connected, which learn global patterns, are used for final classification purpose. The trainable weights in these layers help to extract the important features of input data. The output of CNN network for classification corresponds to the predicted class. The general structure of CNN model is shown in Figure1.

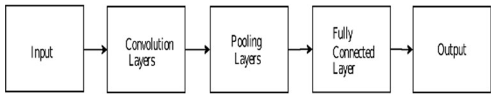

Figure 1. General CNN architecture

The convolution layer learns local patterns in the input data using convolution operation with filters of fixed size. For an input image U having dimension $(m \times m)$ and a filter F having dimension $(f \times f)$, the convolution is denoted by Eq. 1:

$$C = U * F, \tag{1}$$

here is C the convolution map or feature map of dimension c × c, where

$$C = \left(m - f + \frac{2p}{s}\right) + 1. \tag{2}$$

In Eq. 2, p is the padding and s is the stride. Convolution operation between input and the kernel or filter results in feature map. Local connectivity and parameter sharing is obtained with these filters used in convolution operation. Here Rectified Linear Unit (ReLu) is used as activation function after convolution. So,

$$C_{nl} = \mathrm{f(C)}, \tag{3}$$

here *Cnl* is the result of applying ReLu nonlinear activation function on the convolution output. CNNs use a set of filters in a layer and each filter can encode specific features of input. The output of each filter is called activation map or response map. The main aim of max-pooling layers after convolution layers is to downsample feature maps, so that a compressed feature vector is obtained. Kernel size and stride are the main parameters in maxpooling layer. It extracts a window from the feature map and outputs the maximum value of each channel using a max tensor operation. This layer also helps to reduce the number of response map coefficients to process and helps to avoid overfitting. This layer introduces a spatial filter hierarchy, so that the next convolution layer will look at next higher window size. The Fully Connected layer is used for classification. This layer forms local connectivity between pixels and neurons. This final dense layer is connected to a Softmax activation function. The output of a fully connected dense layer after applying a nonlinear activation function is shown in Eq. 4.

$$y^i = f(z^i), \tag{4}$$

where z^i given by Eq. 5:

$$z^i = z^i y^{i-1} + b^i, \tag{5}$$

here w and b are the weight and bias of the network respectively.

B. OVERFITTING AND REGULARIZATION

Overfitting occurs when the model learns most of the features in the training data to the extent that it fails to predict correctly on new data. Large number of learnable parameters is the primary cause of overfitting. Having huge number of model parameters makes the model to learn unwanted data distribution from the training set. Overfitting causes the networks to learn patterns specific to the training set only and not with other data. A good fit is obtained with training data, but it cannot generalize well on data that is not seen during training.

Regularization is a method used to improve the model performance on unseen data and thereby avoiding overfitting. Regularization makes some changes in the learning algorithm. Some amount of randomness is introduced into the learnable parameters to circumvent overfitting. $L1$ regularization, $L2$ regularization, dropout methods and early stopping are the main regularization techniques to avoid overfitting. Data augmentation is another method to overcome overfitting.

In calculating the cost function of a model, some particular parameters are added to the loss function, so that some weights are made negligibly small. This helps to minimize the number of model parameters that can cause overfitting. In $L1$ regularization technique, the absolute value of the network weights is added to the loss function as given below.

$$J = \sum_{i=1}^{n}[y_i - \sum_{j=1}^{m} x_{ij} w_j]^2 + r \sum_{j=1}^{m} |w_j|. \tag{6}$$

In Eq. 6, loss function is represented as the first squared term and absolute value of the weights, which is L1 regularization, is the second term. Here r is a control parameter, which is used for adjusting the fitness of the model. Since L1 regularization is computationally more complex, it is rarely used. L2 regularization is computationally more efficient. It adds a squared value of all weights in the network as a regularizing factor to the loss function, as given in Eq. 7.

$$J = \sum_{i=1}^{n}[y_i - \sum_{j=1}^{m} x_{ij} w_j]^2 + r \sum_{j=1}^{m} {w_j}^2. \tag{7}$$

Dropout is another method to handle overfitting, by deleting some weights in the network randomly. Early stopping helps to understand when to stop training process. The validation loss is also monitored with the training loss. Both losses decrease with training epochs. After an optimum point, validation loss curve will start to increase due to overfitting. Early stopping is used to stop training, when this optimum point is crossed.

III. METHODOLOGY

The methodology of the work is shown in Fig. 2. Since all CNN models are data centric, a large dataset needs to be created for training. The dataset is a mixture of synthetic and the samples from actual receipts. The synthetic data is generated from the windows true type font files since most receipts are made with the help of windows machines. The real dataset is created by first scanning many receipts and segmenting out the letters and grouping them into 62 classes (26 uppercase, 26 lowercase alphabets and 10 numbers). The second step is the creation of a CNN model for letter recognition. The third step is the training and cross validation of the model. The matured model is then deployed to test actual data and its performance is validated. This methodology is implemented with the experimental steps in the next section.

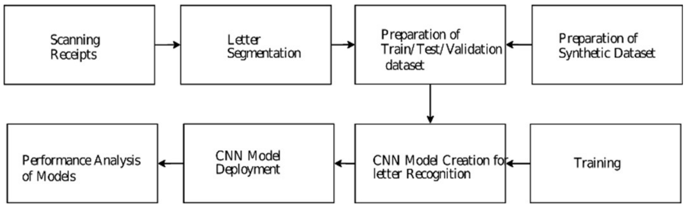

Figure 2. Methodology

IV. EXPERIMENT

Experimental setup for training and testing CNN model is shown in Fig. 4. The various steps are:

1. Preparation of a dedicated dataset for training neural network which contains both handwritten and printed characters.
2. Data augmentation.
3. Development of a CNN model for recognizing letter data from receipts.
4. Hyperparameters tuning to get optimum CNN model performance
5. Different regularization and optimization methods to find the best CNN model.

A. LETTER IMAGE DATA GENERATION

Both synthetic and original (including handwritten) letter images are used for training the CNN model. Letters in receipts usually use Windows fonts with letters of different size. By extracting Windows true type (.ttf) files, it is possible to create letter images of all lowercase and uppercase English characters and numbers. The size of each synthetic image generated is 60×40. These letter images belong to 62 categories and are stored in respective folders. Different receipts are scanned and letters are extracted using python program and appended to the synthetic letter dataset. A total of 60000 letter images are taken for training, testing and validation of the CNN model.

B. PRE-PROCESSING AND DATA AUGMENTATION

The whole 60000letter images are separated into training data set, test data set and validation data set in the ratio 80 : 10 : 10. Each 62 training classes contain 742 samples each. Data augmentation methods help to improve the quality of training data used and avoid overfitting. This can also prevent positional bias. Keras image data generator is used for implementing data augmentation. Rotation, width shift, height shift, rescaling are done on training set. A batch size of 32 is used. The input image is resized to 128 × 128 and interpolation is done. Data is shuffled before feeding it to the neural network to get effective training.

C. DEVELOPMENT OF CNN MODELS

The CNN structure for receipt letter data classification is shown in Fig. 3. The first block represents the input data to the CNN network. Here input data is the different letter images each of size 128 × 128 × 3 obtained from scanned documents and windows (.ttf) files. This input data is pre-processed for noise removal and normalization. The input dimension is a 4D vector with batch size of 32. Convolutional neural network for this 62 class classification of receipt letter images is coded in python using Keras and Tensorflow deep learning library. The input to CNN is letter images of size 128 × 128 × 3, with a batch size of 32. First convolution layer consists of 32 filters, each of size 3 × 3. It is followed by nonlinear activation ReLu and a maxpooling layer of size 2 × 2. This is followed by three more convolution layers with 64, 128 and 128 filters respectively. Each of these layers are followed by nonlinear activation ReLu and a maxpooling layer of size 2 × 2. After flattening, a dense layer of size 512 neurons is connected. Its output is passed to a L2 regularization unit, followed by nonlinear activation ReLu. Then a dense layer of size 62 with softmax activation is used to classify the letter images into 62 classes.

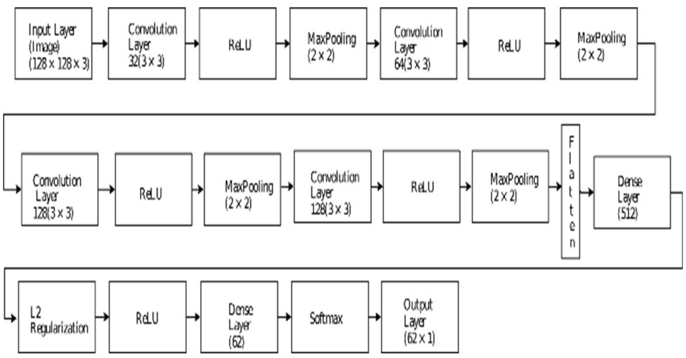

Figure 3. Proposed CNN architecture

D. TRAINING, TESTING AND CROSS VALIDATION

Training is done on NVIDIA Tesla k20M GPU hardware. Early stopping is included in training the CNN in order to avoid overfitting and find out the correct number of training iterations, so that optimum weights are used for predicting the class of images. Hyperparameters like number of filters in each convolution layer, number of maxpooling layer, and choice of activation function are tuned to get better performance. Cost function is minimized in each iteration by updating weights by backpropagation through an optimizer. Adam optimizer and RMS optimizer are considered for analyzing the performance of the model.

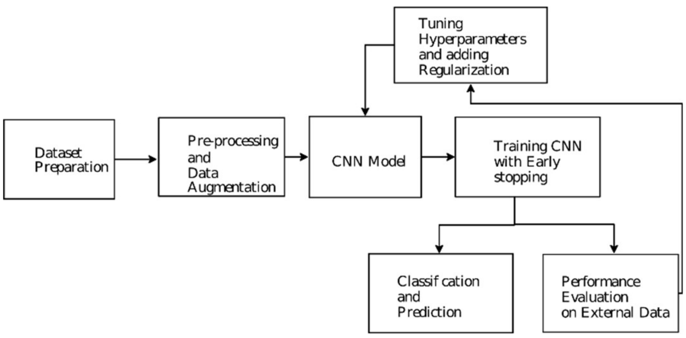

Figure 4. Experimental setup for training and testing CNN model

The pre-processed and augmented data is given to the input of CNN model for training. During the training process, overfitting is avoided by using early stopping. Then performance evaluation is done on external data set and the hyper parameters are tuned to obtain optimum accuracy with the model. Also regularization techniques are added to overcome fitting problems and thereby improve the model performance. Various optimization algorithms are considered to increase the performance.

E. PERFORMANCE COMPARISON

The steps followed in the performance comparison of models and the selection of the best model are shown in Fig. 5. This section includes receipt image data generation, pre-processing and data augmentation, development of CNN models and performance comparison. Different CNN models are considered to get better prediction accuracy on test/external data set. CNN model with $L2$ regularization, CNN model with dropout, CNN model with dropout and $L2$ regularization were considered. Performance is calculated in terms of training accuracy, test accuracy and validation accuracy. $F1$ Score, precision, recall and confusion matrix are also determined on test dataset.

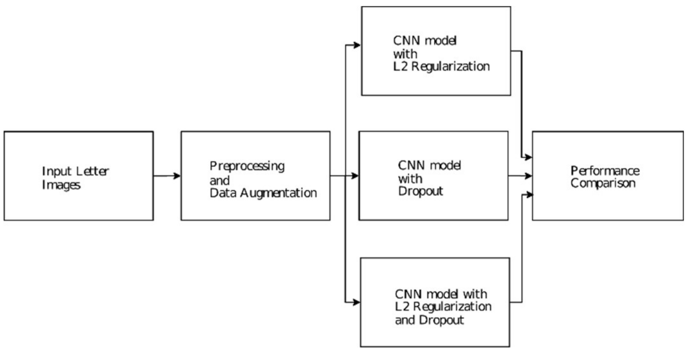

Figure 5. Experimental setup for performance comparison

V. RESULTS AND ANALYSIS

CNN model is fed with all 62 class of input letter images for training. Left shift, right shift, rotation, width shift, height shift, rescaling are done on input data as a part of data augmentation. Results after applying shuffling to the training data are shown in Fig. 6. Another set of input data after applying data shuffling is shown in Figure 7. The CNN model with L2 regularization and early stopping shows better accuracy compared to other methods. Training and validation accuracy of CNN model with L2 regularization is shown in Fig. 8.

Figure 6. Training data set1 after applying data shuffling

A. COMPARISON OF CNN MODELS

To evaluate the performance of the CNN models, accuracy value(A), precision value(P), Recall(R) value and F1 score(F) are calculated for test dataset as shown in Table 1. Both macro average(M.A) and weighted average (W.A) values are considered for precision value(P), Recall(R) value and F1 score(F) . It is evident that CNN with L2 regularization and RMS optimizer gives 92% test accuracy for external dataset. Also, all these values are higher for this model compared to other CNN models. Accuracy(A) is measure used to indicate the model performance in considering all classes. It is the ratio of the number of predictions that are correct to the total number of predictions:

$$\boldsymbol{A} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}}. \tag{8}$$

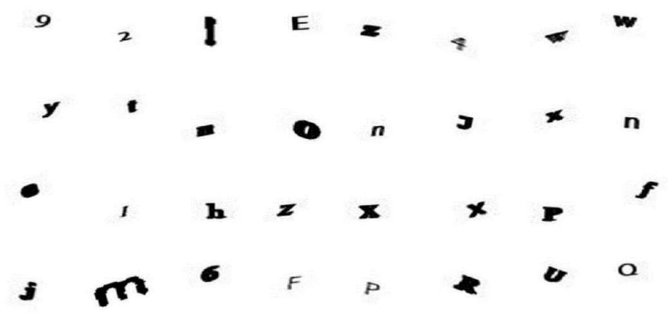

Figure 7. Training data set2 after applying data shuffling

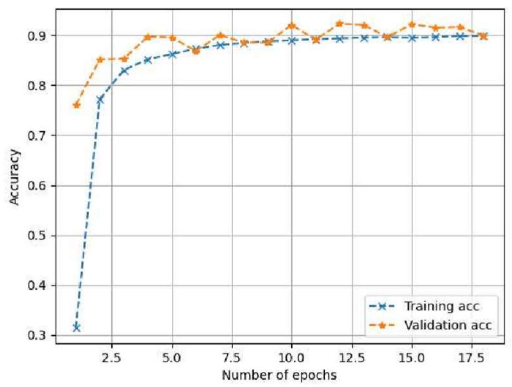

Figure 8. Training and validation accuracy for CNN with L2 regularization

Table 1. Comparison of different CNN models

Model	Opti-mizer	A %	P (M.A)	R (M.A)	*F* (M.A)	P (W.A)	R (W.A)	*F* (W.A)
CNN with L2 regularization	RMS	92	92	91	92	92	92	92
CNN with L2 regularization	ADAM	90	90	90	90	90	90	90
CNN with dropout and *L2* regularization	RMS	90	90	89	90	90	90	90
CNN with dropout	RMS	90	91	90	90	91	90	91
CNN	RMS	91	92	91	91	91	92	91

Precision(P) determines how accurately a model can classify a data as true positive. It is the ratio between the number of data samples (letter images) that are correctly classified as positive to the total number of data samples that are classified as positive:

$$P = \frac{TP}{TP+FP}. \tag{9}$$

Recall(R) is a measure that determines the model's capacity to detect Positive data samples. If recall score is high, it shows that the model detects more positive samples. The higher the recall, the more positive samples are detected. It is the ratio of true positive data to the total number of positive data samples:

$$R = \frac{TP}{TP+FN}. \tag{10}$$

F1 score is a method of combining precision and recall of the model. It is the harmonic mean of Recall and Precision values:

$$F1\, SCORE = \frac{2PR}{P+R}, \tag{11}$$

Here, TP denotes true positive cases, FP denotes false positive cases, FN is denotes false negative cases and TN denotes true negative cases. Training accuracy, validation accuracy and test accuracy are calculated for all five types of CNN models. A comparison is shown in Table 2. CNN with L2 regularization has a training accuracy of 92.2% and testing accuracy of 91.6%, with rms optimizer. It reduces to 90.5% and 89.1% respectively for adam optimizer. CNN with dropout, CNN with dropout and L2 regularization, normal CNN are having lower training and test accuracy compared to CNN with L2 regularization.

Table 2. Comparison of accuracies for different CNN models

Model	Optimizer	Training A %	Validatin A%	Test A %
CNN with L2 regularization	RMS	92.2	91.4	91.6
CNN with L2 regularization	ADAM	90.5	90.2	89.1
CNN with dropout and *L2* regularization	RMS	90.4	89.7	89.7
CNN with dropout	RMS	91.4	90.6	90.7
CNN	RMS	92.2	91.2	91.4

Confusion Matrix helps to find more details about the model performance. The confusion matrix of the test dataset for CNN model with L2 regularization is shown in Fig. 9. The predicted values and the actual values are shown in X and Y axis respectively. So this is a 62×62 matrix. It is evident from the Fig. 9 that high value appears in the diagonal line. This shows that model is correctly classifying the 62 classes with high accuracy.

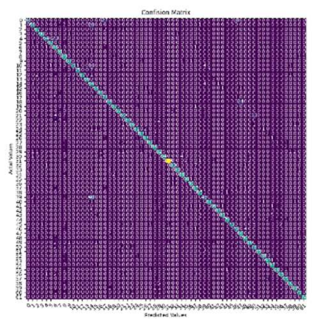

Figure 9. Confusion Matrix

The model performance in noisy conditions is also tested. Prediction error for both Gaussian noise and impulse noise is shown in Figure 10. Prediction error is less for letters corrupted with impulsive noise compared to Gaussian noise. It is evident from this graph, that the model predicts better even in the presence of 10% of both noises. Prediction error linearly increases when percentage of impulsive noise is increased from 0% to 30%. But for letters corrupted with Gaussian noise, the prediction error increases drastically after 10% of noise.

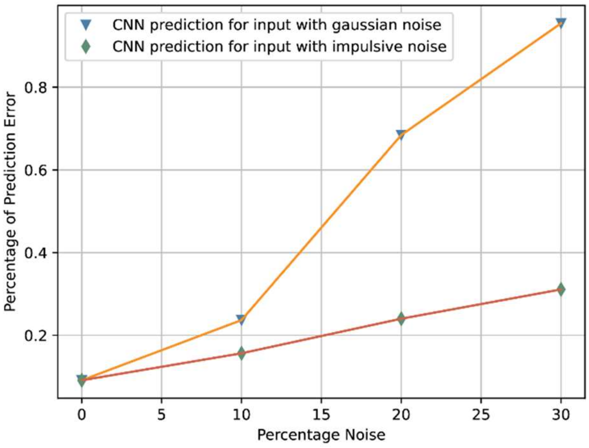

Figure 10. Prediction error for different noise levels

VI. CONCLUSIONS

The proposed CNN model for receipt letter data classification for 62 categories of letters is observed to have good accuracy on test dataset. Hyper parameters of the model are tuned to get optimum performance. The CNN models with L2 regularization, drop put, L2 regularization with drop out are trained and tested. CNN model with L2 regularization is having 92.2% train accuracy and 91.6% test accuracy, which shows better performance compared to other CNN models. Test accuracy is improved with RMS optimizer, when compared to ADAM optimizer. Early stopping was used in the training process to avoid overfitting. Accuracy value, precision value, recall value and F1 score of all the models are calculated and compared. CNN with L2 regularization shows macro average precision of 92%, recall of 91% and F1 score of 92%. In order to observe the predicted classes, confusion matrix is plotted. Future research can focus on text detection from receipts based on this character recognition.

References

[1] K. Y. Wong, R. G. Casey, F. M. Wahl, "Document analysis system," *IBM Journal of Research and Development*, vol. 26, issue 6, pp. 647–656, 1982. https://doi.org/10.1147/rd.266.0647.

[2] J. Memon, M. Sami, R. A. Khan, M. Uddin, "Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR), *IEEE Access*, vol. 8, pp. 142642–142668, 2020. https://doi.org/10.1109/ACCESS.2020.3012542.

[3] S. Mori, C. Suen, K. Yamamoto, "Historical review of OCR research and development," *Proceedings of the IEEE*, vol. 80, issue 7, pp. 1029–1058, 1992. https://doi.org/10.1109/5.156468.

[4] T. K. Ho, G. Nagy, "OCR with no shape training," *Proceedings of the 15th International Conference on Pattern Recognition. ICPR-2000*, vol. 4, 2000, pp. 27–30. https://doi.org/10.1109/ICPR.2000.902858.

[5] G. Nagy, "At the frontiers of OCR," *Proceedings of the IEEE*, vol. 80, issue 7, pp. 1093–1100, 1992. https://doi.org/10.1109/5.156472.

[6] V. Bharath, N. S. Rani, "A font style classification system for English OCR," *Proceedings of the 2017 IEEE International Conference on Intelligent Computing and Control (I2C2)*, 2017, pp. 1–5. https://doi.org/10.1109/I2C2.2017.8321962.

[7] R. Hoch, "Using IR techniques for text classification in document analysis," *Proceedings of the SIGIR'94*, Springer, 1994, pp. 31–40. https://doi.org/10.1007/978-1-4471-2099-5_4.

[8] B. B. Chaudhuri, U. Pal, "A complete printed Bangla OCR system," *Pattern Recognition*, vol. 31, issue 5, pp. 531–549, 1998. https://doi.org/10.1016/S0031-3203(97)00078-2.

[9] A. Shrivastava, I. Jaggi, S. Gupta, D. Gupta, "Handwritten digit recognition using machine learning: A review," *Proceedings of the 2019 IEEE 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC)*, 2019, pp. 322–326. https://doi.org/10.1109/PEEIC47157.2019.8976601.

[10] C. Zhang, Z. Zhou, L. Lin, "Handwritten digit recognition based on convolutional neural network," *Proceedings of the 2020 IEEE Chinese Automation Congress (CAC)*, 2020, pp. 7384–7388. https://doi.org/10.1109/CAC51589.2020.9326781.

[11] J. Li, G. Sun, L. Yi, Q. Cao, F. Liang, Y. Sun, "Handwritten digit recognition system based on convolutional neural network," *Proceedings of the 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, 2020, pp. 739–742. https://doi.org/10.1109/AEECA49918.2020.9213619.

[12] S. S. Rajput, Y. Choi, "Handwritten digit recognition using convolution neural networks," *Proceedings of the 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, 2022, pp. 0163–0168. https://doi.org/10.1109/CCWC54503.2022.9720854.

[13] J. Hu, M. K. Brown, W. Turin, "HMM based online handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, issue 10, pp. 1039–1045, 1996. https://doi.org/10.1109/34.541414.

[14] C. Bahlmann, B. Haasdonk, H. Burkhardt, "Online handwriting recognition with support vector machines – A kernel approach," *Proceedings of the Eighth IEEE International Workshop on Frontiers in Handwriting Recognition*, 2002, pp. 49–54.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, issue 6, pp. 84-90, 2017. https://doi.org/10.1145/3065386.

[16] N. K. Manaswi, N. K. Manaswi, S. John, *Deep Learning with Applications using Python*, Springer, 2018. https://doi.org/10.1007/978-1-4842-3516-4.

[17] K. A. Hamad, K. Mehmet, "A detailed analysis of optical character recognition technology," *International Journal of Applied Mathematics Electronics and Computers*, vol. 4, Special Issue 1, pp. 244–249, 2016. https://doi.org/10.18100/ijamec.270374.

[18] S. Khare, J. Singh, "Handwritten Devanagari character recognition system: A review," *International Journal of Computer Applications*, vol. 121, pp. 9, 2015. https://doi.org/10.5120/21566-4600.

[19] T. Ashwin, P. Sastry, "A font and size-independent OCR system for printed Kannada documents using support vector machines," *Sadhana*, vol. 27, issue 1, pp. 35–58, 2002. https://doi.org/10.1007/BF02703311.

[20] M. Avadesh, N. Goyal, "Optical character recognition for Sanskrit using convolution neural networks," *Proceedings of the 2018 13th IEEE IAPR International Workshop on Document Analysis Systems (DAS)*, 2018, pp. 447–452. https://doi.org/10.1109/DAS.2018.50.

[21] S. Joshi, N. Khanna, "Single classifier-based passive system for source printer classification using local texture features," *IEEE Transactions on Information Forensics and Security*, vol. 13, issue 7, pp. 1603–1614, 2018. https://doi.org/10.1109/TIFS.2017.2779441.

[22] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press, 2016.

[23] C. Tensmeyer, T. Martinez, "Analysis of convolutional neural networks for document image classification," *Proceedings of the 2017 IEEE 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, 2017, pp. 388–393. https://doi.org/10.1109/ICDAR.2017.71.

[24] C.-J. Lin, Y.-C. Liu, C.-L. Lee, "Automatic receipt recognition system based on artificial intelligence technology," Applied Sciences, vol. 12, issue 2, 853, 2022. https://doi.org/10.3390/app12020853

[25] L. Chen, S. Wang, W. Fan, J. Sun, S. Naoi, "Beyond human recognition: A CNN-based framework for handwritten character recognition," *Proceedings of the 2015 IEEE 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, 2015, pp. 695–699. https://doi.org/10.1109/ACPR.2015.7486592.

[26] A. Yue, *Automated Receipt Image Identification Cropping and Parsing*, Princeton. Edu, 2020, pp 1-8.

[27] P. Y. Simard, D. Steinkraus, J. C. Platt, et al., "Best practices for convolutional neural networks applied to visual document analysis," *ICDAR*, vol. 3, pp. 122-132, 2003.

***DESIREE JUBY VINCENT** received B Tech degree in Electronics and Communication from Cochin University of Science and Technology in 2010 and M Tech degree in Electronics (VLSI and Embedded Systems as specialization) from Cochin University of Science and Technology in 2013. She is currently pursuing research in Non-linear signal and image processing in Kerala Technological University. She also worked as Assistant Professor in Department of Electronics and Communication in Bishop Jerome Institute, Kollam. Her research interest is in designing Machine learning and Deep learning models for signal and image processing.*

***HARI V S** did B Tech in Electronics and Communication from University of Kerala and M. Tech with specialization in Communication Systems from IIT, Madras. He earned PhD from School of Engineering, Cochin University of Science and Technology. His interests are in Polynomial Signal Processing, Machine learning and Digital Communication Systems. He works as Associate Professor with Department of Electronics, College of Engineering, Chengannur, Kerala, India. He is also the Director of the company Rand Walk Research and Solutions Pte. Ltd.*