Evaluation of CNN Models on the Classification of the Modified MNIST Dataset

Yimeng Hu **Long-Quan Bach** **Sichen Wan**

Abstract

This project evaluates the performance of several convolutional neural network (CNN) models on classifying the Modified MNIST Dataset. Specifically, we built a 9-layer CNN model, which serves as a baseline model to compare with the state of the art, including VGGNet, ResNet and the ensemble of several ResNet models. These models are modified accordingly to fit our analysis. We found that the ensemble of ResNets achieves the highest prediction accuracy of 97.6%, closely followed by the ResNet model of 97.3% and VGGNet of 96.9% accuracy, which is a significant improvement over the baseline of 91.7% accuracy. We discover that by increasing the number of layers of our baseline model to 13, the accuracy increases to 96.3%, which is close to the result obtained from VGGNet and ResNet.

1 Introduction

1.1 Project Task

The task of this project is to perform an image analysis prediction challenge on Modified MINIST Dataset. This dataset contains 50,000 images with associated labels as the training set. Each image contains three digits, and the goal is to predict the digit in the image with the highest numeric value on the held-out test set containing 10,000 images. To achieve this, we investigated several Convolutional Neural Network (CNN) models, which include a simple 9-layer CNN model we built as a baseline, a modified VGGNet, a modified ResNet, and an ensemble model consisted of four ResNets. VGGNet and ResNet are the state-of-the-art models which have been proven to achieve great success in large-scale image classification tasks. Before the implementation of these models, we explored several preprocessing techniques, including normalization, noise reduction, and training set transformations such as random rotation and random shift to expand the size of our training set. Finally, we applied several regularization and optimization strategies to improve the performance of our models.

1.2 Important Findings

We found that the ensemble of the models outperforms other models, achieving a prediction accuracy of 97.6%. This is closely followed by the ResNet model of 97.3% and VGGNet of 96.9% accuracy. The accuracy achieved by these state of the art is a significant improvement over the baseline of 91.7% accuracy. However, adding 4 additional convolution layers to our baseline model increased the accuracy to 96.3%, which is close to the result obtained from VGGNet and ResNet. This proved that increasing the convolution layers of a CNN model can significantly improve its performance.

2 Related Work

The task at hand is image analysis prediction which is an expansive research field. Our work is very similar to the classification of the MNIST dataset [3], classifying handwritten numerical digits. There are plenty of studies on this dataset, including the evaluation of CNN on classification [14]. The idea of individually extracting digits and classifying them based on their characters was a viable option for our task, and was also a very popular research field as well. This idea is otherwise known as Optical Character Recognition (OCR). Vehicle license plate recognition [4] explores techniques of extracting the license plate numbers from a car and classifying their digits and letters based on a separate

database. The popular Street View House Number (SVHN) dataset consists of images of house numbers collected from Google Street View images. Digits that appear on a variety of backgrounds are extracted using contours and are classified individually using a MNIST-like database consisting of digits 0-9 [5].It was also possible to skip any OCR and create a deep CNN to handle our task, very much like the approach most state-of-the-art results for the ImageNet challenge(Figure 1). We decided to focus on creating deep convolutional neural networks to handle the classifications of our Modified MNIST and drew most of our influence from model performances on ImageNet. This resulted in the exploration of VGGNet[12] and ResNet[6].

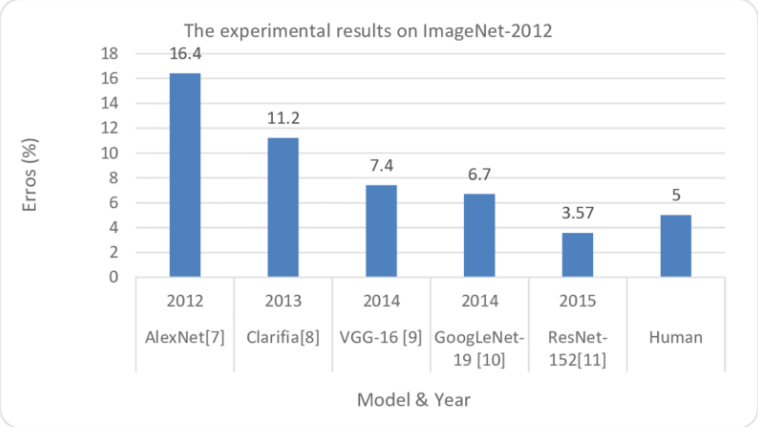


Figure 1: Performances of different deep learning models on ImageNet. Image from Alom et. al. [1]

3 Dataset

3.1 Dataset Description

The Modified MNIST dataset was constructed based upon the original MNIST dataset [3]. The original MNIST dataset contains handwritten numeric digits with labels 0-9. In Modified MNIST, 3 handwritten digits are randomly placed onto a 128x128 pixel image accompanied with background noise. These images are stored as a matrix representing grey-scale pixel intensity values ranging from 0 (black) to 255 (white). The Modified MNIST dataset contains 50,000 images. In our experiments, we split this dataset into 40,000 training samples and 10,000 validation samples. In the kaggle competition, the prediction is made on a held-out dataset of 10,000 images.

3.2 Preprocessing and Data Cleaning

Due to the concern that our neural networks may have trouble recognizing digits because of the noisy background, we decided to remove the background noise to extract only the digits. Fortunately, the digits were significantly darker than the rest of the image, making it slightly easier for us to extract them. There have been plenty of research dedicated to optical character recognition (OCR) with very similar tasks. We decided to take a similar approach for vehicle license-plate recognition used by Duan et. al. [4].

For each image, we applied a median filter, a non-linear filtering technique to blur the noises in the image. Median filtering was chosen to help preserve some of the sensitive edges on digits opposed to the more popular gaussian filter [2]. A threshold was applied to the image to remove this background noise. Then we used erosion to remove any small, existing noise that remained in the background followed by a dilation to further enhance the sizes of the remaining digits. These steps can be seen chronologically in Figure 2. Besides, we normalized the data to ensure the range of the value in the matrix is from 0 to 1.

3.3 Dataset Augmentation

Data augmentation is a practical technique that can be used to expand the size of our training set to help our models generalize better. It has been proven before that training data expansion is key to reducing error rates in model predictions for CNNs [11].

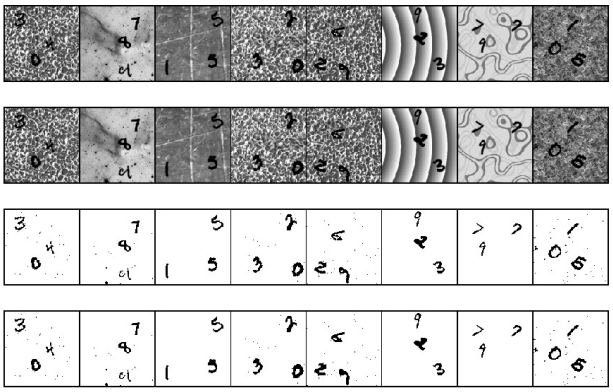


Figure 2: Noise reduction on the Modified MNIST dataset using median filter, applying a threshold, and morphological transformation.

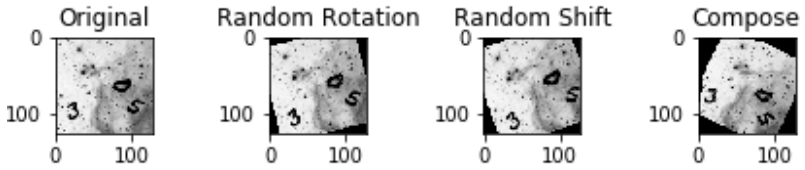


Figure 3: Training set expansion using the compose for random rotation and random shift.

Therefore, we performed transformations on the training set. The transformation we used composes of a random rotation followed by a random shift. A small parameter for the angle rotation and image shift was chosen to preserve all the numbers on the image so that none are accidentally cut out (Figure 4). These transformations were be applied to the validation or the test set.

4 Proposed Approach

4.1 Baseline CNN

CNN is one of the main architectures for image recognition/classification tasks. It is a deep and multi-layered neural networks whose convolution layers alternate with pooling and fully connected layers. The ability of extracting information on different regions of the images makes CNN an obvious candidate for our task[10].

CNNs vary in how convolution and subsampling layers are connected. We started off with a very simple architecture implemented using 6 convolution layers and 3 fully connected layers as our baseline model. We tried to improve it by changing the parameters of the layers and by tuning the hyperparameters. An increase in performance was observed with the increase of numbers of convolution layers. The final model consists of 10 convolution layers together with 3 fully connected layers. It is chosen based on accuracy and average loss on validation set. To avoid overfitting and to improve the generalization of deep neural networks, we applied both batch normalization and dropout regularization to our model. Batch normalization increases the stability of the neural network by normalizing the output of a previous activation layer[7]. Dropout is a regularization technique where randomly selected neurons are ignored during training and hence results in a network that is less sensitive to a specific weight of neurons and hence capable of better generalization[13].

4.2 Modified VGGNet

We applied a modified version of VGGNet[12] and we trained the model on our training dataset. VGGNet is a state of the art convolutional neural network (CNN) developed by VGG (Visual Geometry Group) from the University of Oxford. This CNN uses very deep convolutional layers with very small (3×3) convolution filters in all layer. It won second place in ILSVRC (ImageNet Large Scale Visual Recognition Competition) in 2014, with GoogLeNet at the first place. Although it was not the winner, it significantly improved over ZFNet[15], the winner in 2013, and it was

the first year that the deep learning models obtaining the error rate under 10%. There are several options of VGGNet models with 11, 13, 16, and 19 layers respectively. We decided to use the 16-layer VGGNet (VGGNet16) because it was proven to have the lowest error rate in the task of image classification [12]. We modified the VGGNet16 by the following way to fit our analysis. First,we changed the input channel from 3 to 1 since we are using 1 channel images. Second, we changed the output dimension from the original 100 to 10 since we are performing 10-class classification. Third, we added batch normalization layers to the model because it allows the use of a much higher learning rates and a relatively less careful weight initialization, which improves the stability of the model and accelerates the learning[7].

4.3 Modified ResNet

We implemented a version of the ResNet-18 model developed by He et. al. [6] which was able to win first place in the ImageNet challenge back in 2015 and achieve superhuman performance. Some adjustments had to be made to the ResNet model so that it can accept our data set. We changed the number of input channels from 3 to 1, as our images are in grayscale and not RGB, and we reduced the size of the final output layer to 10, resembling the number of labels in our prediction. Finally, our images had to be resized to 224x224 pixels to be accepted as a valid input. The authors concluded in the paper that there is a direct relationship between the number of layers in ResNet and the accuracy rate on ImageNet [6]. We implemented ResNet-34 and ResNet-50 to test if this is true for our dataset and performed the same modifications mentioned above for each model.

4.4 Ensemble

We will use ensemble learning as our final method. It has been proved in many image classification tasks that a combination of several classifiers will always perform better than one [9]. In this case, we decided to ensemble 4 ResNets. Instead of bagging training samples, we used the training data set expansion technique discussed in section 3.3 to have each model be trained with augmented data. This allows each model to be trained with the maximum amount of inputs while limiting the risk of overfitting. Ensemble predictions will follow a majority voting system.

4.5 Optimization Strategies

The optimization strategies we applied are Adam optimizer [8] and learning rate scheduler. Adam is an algorithm for first-order gradient-based optimization of stochastic objective functions. This optimizer computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients. Learning rate scheduler enables the adjustment the learning rate based on the number of epochs.

5 Results

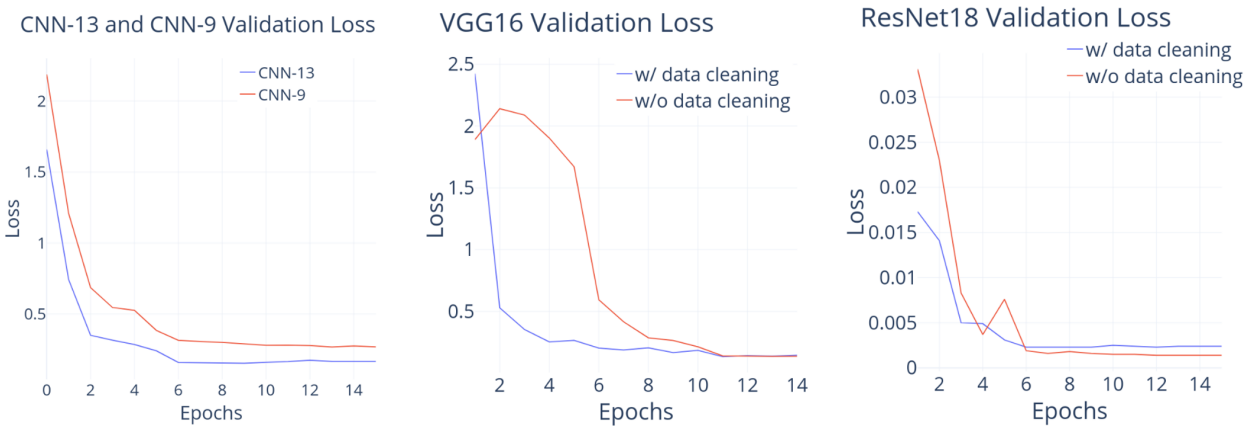
This section covers the results of the above mentioned learning algorithms. In all cases, we evaluated the cross-entropy loss in the validation set to find the best model.

Table 1. demonstrates both the competition accuracy and the validation accuracy measured during the training session. We can see that the ensemble model achieved the best performance of 97.6% accuracy, followed by the modified ResNet and VGGNet model of 97.3% and 96.9% accuracy respectively. The accuracy achieved by these models is of significant improvement over our baseline model with 91.7% accuracy. We improved the model with 4 additional convolution layers, which achieved 96.3% accuracy on the validation set. This close to the accuracy we obtained from the state-of-the-art models we used.

Learner	**Competition Accuracy %**	**Validation Accuracy %**
CNN-9	-	91.7
CNN-13	-	96.3
VGGNet	96.9	97.0
ResNet18	97.3	97.6
ResNet50	-	96.3
Ensemble	97.6	-

Table 1: Learner Accuracy

Figure 4-a) compares the performance of the two baseline CNN models on their validation loss over epochs in Figure 4-a). We can see that the validation loss converges quickly until it started plateauing at around 10 epochs.



a) 9-Layer vs. 13-Layer CNN Model Loss Over Epoch

b) VGGNet Model Loss Over Epoch With and Without Noise Reduction.

c) ResNet18 Model Loss Over Epoch With and Without Noise Reduction

Figure 4: Plot of Validation Losses for Different Deep Learning Models.

As for Modified VGGNet (Figure 4-b), it achieved a good performance as expected. Noise reduction does not seem to be the key to obtaining a good accuracy rate as demonstrated in Figure 4-b), since the loss eventually converges. However, evidently, it converges much faster after denoising. ResNet(Figure 4-c) has a similar converging pattern to VGGNet, whereas the noise reduction increased the loss.

6 Discussion and Conclusion

Despite the simplicity of our baseline models, it achieved a accuracy close enough to our model state-of-the art models. We concluded from this that the number of convolution layers has a great influence on the accuracy of the model. Moreover, we discovered that image prepossessing and parameter initialization are the keys to obtaining a decent accuracy rates within a shorter time period. We also found that though denoising the image generally speeds up the training period as the loss function converges much faster, it does not have an obvious effect on the accuracy rates. Nevertheless, we observed an improvement of the performance with the data augmentation. Besides, we explored various regularization and optimization techniques, which reduces the chance of any undesirable divergent behavior in the loss function.

We recognized that there are certain constraints of this project. First being that due to the time constraints, we mostly used the default hyperparemeters of the optimizers and regularizers. This can be further improved by gird search. Moreover, we only ensembled the ResNet models. Exploration of other ensembles could be done for future enhancement.

Earlier in the project we were having trouble implementing OCR and have a program manually extract digits from images. In theory, if we train a model on the MNIST dataset to recognize these digits independently we can use the same model to classify the extracted digits. However, due to the complexity of the work, we decided to deviate from this idea. For future work, we hope to return to this approach and implement a smarter algorithm to extract these digits.

7 Statement of Contribution

Yimeng implemented the two baseline CNN models Sichen was in charge of implementing the VGGNet Long-Quan explored various Data preprocessing techinques and implemented ResNet and the ensembles.

References

[1] Md Zahangir Alom, Tarek M Taha, Christopher Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Brian C Van Esesn, Abdul A S Awwal, and Vijayan K Asari. The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*, 2018.

[2] Ery Arias-Castro, David L Donoho, et al. Does median filtering truly preserve edges better than linear filtering? *The Annals of Statistics*, 37(3):1172–1206, 2009.

[3] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[4] Tran Duc Duan, TL Hong Du, Tran Vinh Phuoc, and Nguyen Viet Hoang. Building an automatic vehicle license plate recognition system. In *Proc. Int. Conf. Comput. Sci. RIVF*, number 1, pages 59–63, 2005.

[5] Ian J Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks. *arXiv preprint arXiv:1312.6082*, 2013.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. pages 448–456, 2015.

[8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[9] Louisa Lam and SY Suen. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 27(5):553–568, 1997.

[10] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[11] Vadim V Romanuke. Training data expansion and boosting of convolutional neural networks for reducing the mnist dataset error rate. 2016.

[12] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[13] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[14] Siham Tabik, Daniel Peralta, Andrés Herrera-Poyatos, and Francisco Herrera. A snapshot of image pre-processing for convolutional neural networks: case study of mnist. *International Journal of Computational Intelligence Systems*, 10(1):555–568, 2017.

[15] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.