Mini-Project 3: Modified MNIST

Shashank Murugesh
McGill University
shashank.murugesh@mail.mcgill.ca
260786369

Anthony Ho
McGill University
anthony.ho@mail.mcgill.ca
260501840

Julien Courbebaisse
McGill University
julien.courbebaisse@mail.mcgill.ca
260614548

***Abstract*—The task at hand for this project was to build a deep learning model to perform an image analysis prediction challenge on Modified MNIST dataset. Different models such as ResNet50, ResNet50 + Random Forest, ResNet50 + LSTM were evaluated based on there accuracy metric. This project was also part of Kaggle Competition on which our team's best accuracy came out to 96.80%. This was achieved using a Resnet50 model.**

I. Introduction

In the past decade, deep learning neural networks have gained popularity. This is because of the innovations in the application of convolutional neural networks to image classification tasks. The most prominent innovations came from the submissions by researchers at ImageNet Large Scale Visual Recognition Challenge, or ILSVRC [1].

This project consisted in a classification task on a test set of 10,000 images. The 50,000 provided training images featured 3 numbers from the famous MNIST dataset of handwritten digits over various backgrounds. The goal for the project was to build a deep learning model to output the greatest of the 3 numbers present on each picture.

The architecture chosen by our team is the ResNet architecture. Residual Neural Networks are at type of CNN and were first introduced in 2015. At the time it was the first model to beat human performance on the ImageNet classification competition. ResNet solves the problem of vanishing gradients by implementing a way to skip over layers using identity blocks. For this project, ResNet seemed like the better option in terms of both performance and ease of implementation. The team also experimented with an ensemble of both ResNet and Random Forest as well as LSTM but both models came up short in comparison with ResNet alone.

II. Related work

Image classification using Deep Learning has gained much popularity since the beginning of the ImageNet Competition. GoogleNet and Resnet50 are both recognized as some of the better models for image classification[2]. Much research has also been done on predicting the handwritten digit database MNIST. It has been found that deep neural networks performed very well at the task [3]. The best know model for predicting the 10,000 image datsaset used an ensemble of 6 CNNs and performed with an error rate of 0.21% [4]. There is also evidence that techniques know as training data expansion can be very helpful in reducing error rates for this task[4].

III. Dataset and setup

The provided dataset for the project is a variation on the famous MNIST dataset. The MNIST dataset (Modified National Institute of Standards and Technology) is a large database of handwritten numbers used to train neural networks. Many different models have been tested on this dataset ranging from SVMs to CNNs. The best recorded attempt at predicting the original MNIST dataset recorded an error rate of 0.21% and used multiple CNNs.

The dataset at hand for this project is a collection of 128 by 128 pixel images which all feature 3 numbers taken from the original MNIST dataset pasted randomly onto a collection of backgrounds. The digits also never overlap in the images and always have very high intensity, which is useful for preprocessing. Indeed the team was able to threshold individual pixels to remove most of the background noise and obtain a clear image containing digits.

IV. Proposed Approach

In order to classify highest digit in the given image data set. We trained our dataset on different deep learning models.
Below are the various models which we used in this project:

A. *Resnet50*

Resnet is one of the most groundbreaking work in the computer vision/deep learning community as it achieved superhuman accuracy in ILSVRC 2015 contest[5]. Resnet50 is the 50 layer deep residual network. The core architecture of ResNet introduces a so-called "identity shortcut connection" that skips one or more layers, as in Fig. 1. This connection helps to transfer back to earlier layers when the gradient vanishes from the weight layer. Thus, it performs better even when the network is deep.

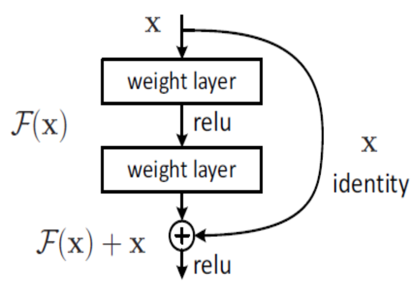

Fig. 1. A Building Block of Residual Network

B. *Ensemble learning with Resnet50 and Random Forest*

This model uses CNN Resnet50 for feature extraction and Random Forest for digit classification. A Resnet50 model is first trained with dataset, and the layers up to the fully connected layer of the trained model (up to the last Max Pooling layer) will be used to extract features from the MNIST images.

The extracted features will be used as the training data for random forest model for max digit classification. The random forest model used in the report is the RandomForestClassifier from scikit-learn ensemble package.

C. *Ensemble learning with Resnet50 and LSTM*

This method is similar to the ensemble model with Resnet50 and random forest, where the model uses Resnet50 for feature extractions and LSTM to learn the hierarchy of the number digits.

The extracted features from Resnet50 will then feed into LSTM for recurrent network training. The LSTM model used in the report is the LSTM layer from keras layers package.

V. Results

Table I provides the compression of three different models on which we trained modified MNIST dateset and we observe that the ResNet model with deep residual 50 layers performs better than other two ensemble learning models. we implemented ResNet using keras library and adapted various image prepossessing technique such as rotation and feature scaling which helped in improving model accuracy.
We trained our model on 40000 images and validated on 10000 images. Fig. 2 Fig. 4 compares training accuracy with validation accuracy of ResNet and ResNet+LSTM model respectively. Fig. 3, Fig. 5 compares training loss with validation loss of ResNet and ResNet+LSTM model respectively.

Model	Validation Accuracy
ResNet50	0.985
ResNet50 + Random Forest	0.972
ResNet50 + LSTM	0.867

TABLE I
Comparison of different models with their mean accuracy

Resnet model with 50 deep residual layers gave us the best validation accuracy score of 98.50%. Thus, we used this model in Kaggle competition.

VI. Conclusion

Some key takeaways from this project are 1) ResNet model performed better than Ensemble model. 2) Image prepossessing technique such as rotation helped the model to be robust and learn better. 3)There was no significant change in accuracy when different optimizer such as adam, adagrad was used. 4) Dropout layer helped model to prevent over fitting.

VII. Statement of Contributions

Table II lists contribution from different group members.

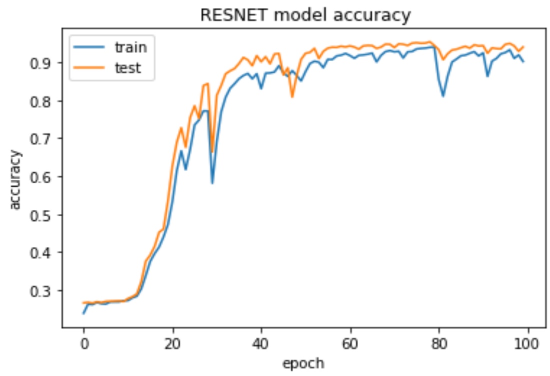

Fig. 2. ResNet50 Training and Validation accuracy

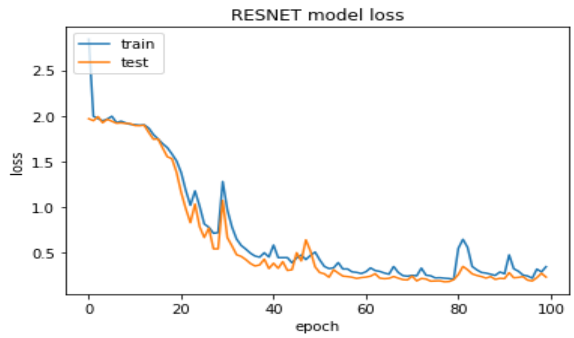

Fig. 3. ResNet50 Training and Validation loss

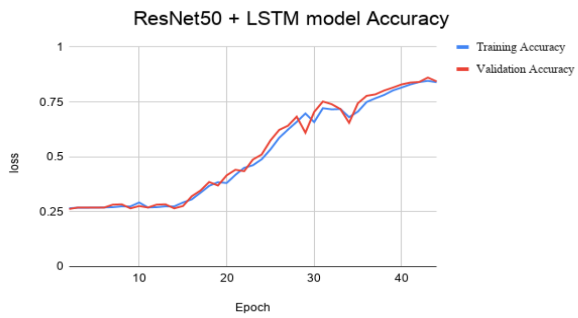

Fig. 4. ResNet50 + LSTM Training and Validation accuracy

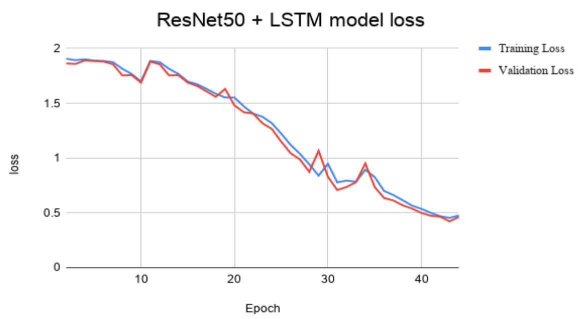

Fig. 5. ResNet50 + LSTM Training and Validation loss

Shashank Murugesh	Anthony Ho
Dataset processing	Dataset processing
Implemented ResNet50	ResNet50 + Random Forest, ResNet50 + LSTM
Running tests	Running tests
Report	Report

Julien Courbebaisse
Dataset processing
Running tests
Report

TABLE II
Statement of contributions

REFERENCES

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[2] A. M. Neha Sharma, Vibhor Jain, "An analysis of convolutional neural networks for image classification," *Procedia Computer Science*, vol. 132, pp. 377–384, 2018.

[3] D. C. U. M. J. Schmidhuber, "Multi-column deep neural networks for image classification," *IDSIA*, 2012.

[4] V. Romanuke, "Training data expansion and boosting of convolutional neural networks for reducing the mnist dataset error rate," *Khmelnitskiy National University*, 2016.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.