Handwritten Characters Recognition Using Neural Network with Out-of-Distribution Detection

Jinghan Jia, *4713-4397*, Ran Dou, *4759-2951*, Sihao Lv, *1888-8483* and Zhengyu Li, *2945-3969*

***Abstract*—With the rapid development of machine learning (ML) and neural networks (NN) theories along with powerful artificial intelligence (AI) toolkit, there are many useful solutions have been creating in order to solve real world AI problems. Handwritten letters recognition is a one of the typical problems in AI and computer vision (CV) field. In this report, our group has figured out handwritten letters in both easy and hard dataset. "Easy" dataset which only includes handwritten letter 'a' and 'b' and "hard" dataset contains 'a', 'b', 'c', 'd', 'h', 'i', 'j', 'k' and unknown characters. Our finding is that each student has their own writing style so that we must disrupt the order of training data in order to equally training and testing.**

***Index Terms*— CNN, deep learning, handwritten letter recognition, machine learning, SVM**

I. Introduction

Our experiment is to determine the recognition accuracy of handwritten letters. The goal on the "easy" dataset is to ensure the recognition accuracy of 'a' and 'b' on the unknown test data could exceed 90 percentage and the goal on "hard" dataset is to obtain the accuracy of 8 letters classes and unknown class as high as possible. Therefore, we used a support vector machine (SVM) and several different NNs to train and test data and we fitted only one model for both easy and hard dataset. Our training dataset contains 6400 letters and each letter is an equal distribution. We used cross validation to train and test the model. For the unknown characters, we used MNIST and EMNIST data as our training data. We designed an SVM classifier to determine the test data is known or unknown. We set the label of data '-1' if it is unknown, and sent the data to NN classifier to set the specific label if it is known data.

Handwritten letter recognition has been a hot topic for a long time. In 1998, Y. LeCun et al came up with two systems for on-line handwriting recognition, and described a Graph Transformer Network (GTN) for recognizing bank check. This algorithm solves the problem that system architecture is influenced by each new input, which is known as credit assignment problem. [4] Recently, due to the improvement of big data computing ability, machine learning has been attracting many researcher's attentions. In 2010, Dewo Nasion et al combined digital image processing algorithm with machine learning. They implemented chain code to represent the characteristic of characters, and as the input data for SVM algorithm and got a good result. [5] In 2017, Zeyi Wen et al proposed Alpha seeding to accelerate cross-validation of SVM algorithm. In addition, Sergey Zagoruyko introduced Wide Residual Networks as an architecture of convolution neural networks in 2016 which was a significant achievement.[1] Besides that, since there were many achievements in English handwritten recognition, many researchers were also researching handwritten recognition in other languages. Ruyu Zhang et al came up with a new method combined both wide residual networks and Center Loss Based Metric Learning in Chinese letter recognition. [6] However, all above algorithms are related to known dataset, instead of out-of-distribution detection. In 2018, Terrance DeVries et al proposed learning confidence to determine the unknown data.

II. Implementation

In order to train the easy dataset, the first step was to process images. The training images were re-sized to 32*32 sizes in our project, and we implemented permutation function used for disrupting the order of images. We used Pytorch and implemented Lenet, Wide Resnet [1] and Alexnet NN to train and validate our model. A weight parameter was set for boosting these three NNs results. The parameter was 1 for Lenet and Alexnet, 2 for Wide Resnet. Eventually, the sum of weighting parameters multiplies the results would be considered as our final result and predicted results would be 1 to 8 corresponds to 'a' to 'k'. The method of our validation here was to equally select 800 data with different labels from our training dataset as our validation data, and we didn't train these randomly selected data at all.

For training the hard dataset, the first step was to download the training data on MNIST and EMNIST. 200 data of each letter of other 18 letters and 280 data of each digital number were randomly selected as our unknown dataset. We did the

This work was supported in part by EEL5840 group 8086k.

Zhengyu Li is with the University of Florida, Gainesville, FL 32608 USA (e-mail: zhengyuli@ufl.edu).

Ran Dou is with the University of Florida, Gainesville, FL 32608 USA (e-mail: dour@ufl.edu).

Jinghan Jia is with the University of Florida, Gainesville, FL 32608 USA (e-mail: jinghan.jia@ufl.edu).

Sihao Lv is with the University of Florida, Gainesville, FL 32608 USA (e-mail: lyusihao@ufl.edu).

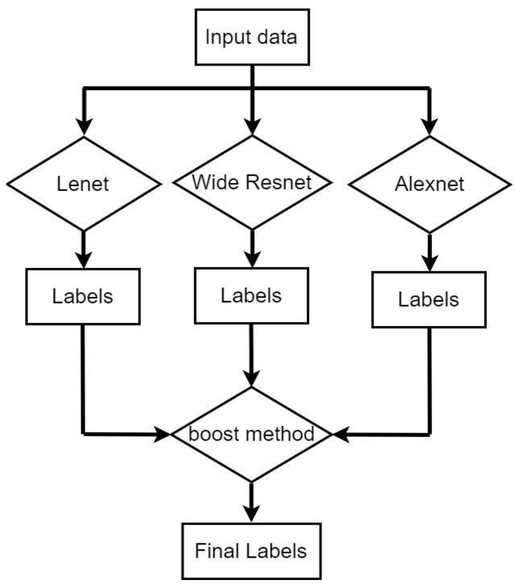

Fig. 1. Flow chart for training and testing Wide Residual Networks model

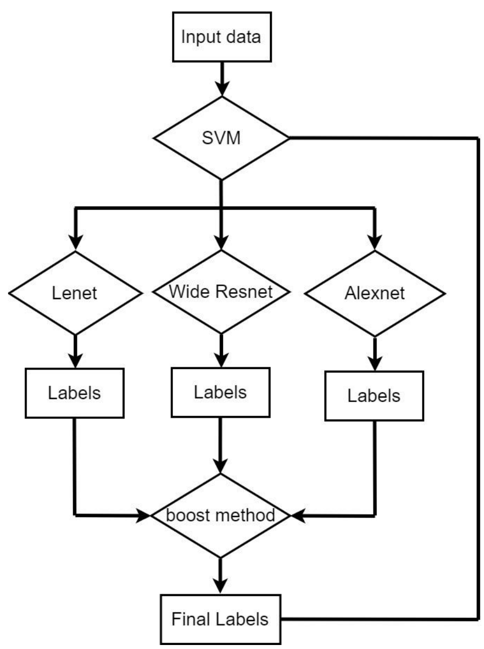

Fig. 2. Flow chart for training and testing Wide Residual Networks model with SVM pre-classifier

same image processing steps to these unknown data as the given dataset. We appended these data to our original dataset and normalized data using StandardScaler function. Then we added an SVM model to classify the known data with label '1' and unknown data with label '-1'. A 10-fold cross validation is used to validate the classifier's performance and avoid overfitting [2]. For those data which predicted label '1' by the SVM model would be sent to the previous three NNs model and would be predicted a final label in that model. The block diagram of our model is shown in figure 1 for processing easy dataset. For the block diagram of hard dataset, we added an SVM classifier unit between input data and three neural networks. If the data is unknown, the final label will be determined immediately; otherwise, data flows to the same model as processing easy dataset. Figure 2 is shown the hard dataset with unknown data.

III. Experiments

The experiments we did in testing NN classifier include randomly select 800 data with equally labels as validation dataset. In other words, we randomly selected the same amount of data for each class in an unordered dataset. What we found here was that randomly select validation data would be better than in order select data. For instance, accuracy of testing first 800 data and training the rest of data is lower than testing randomly selected validate data and training data. Our analysis is that each student has their own writing style and we guess every ten consecutive letters are created by the same student. That is for letter 'a', index 0 to 9 in our training data is created by a student and index 80 to 89 is created by another student. Therefore, the first ten letter 'a' are similar with each other; in other words, they have very similar features. However, the feature of first ten consecutive 'a' will have different features compared with the second ten consecutive 'a' since they were created by different students. Therefore, we should also consider the location of the same class data and try to put letters in each class wrote by each student into both train and validation set, rather than only training or testing letter in one class wrote by a student. The accuracy is around 98 percentage on predicting the label of random validation data.

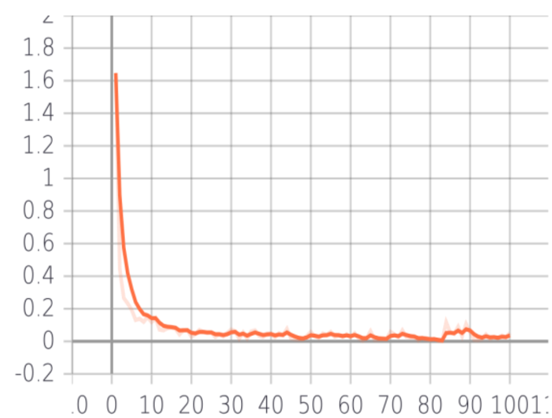

Fig. 3. Alexnet Train Loss

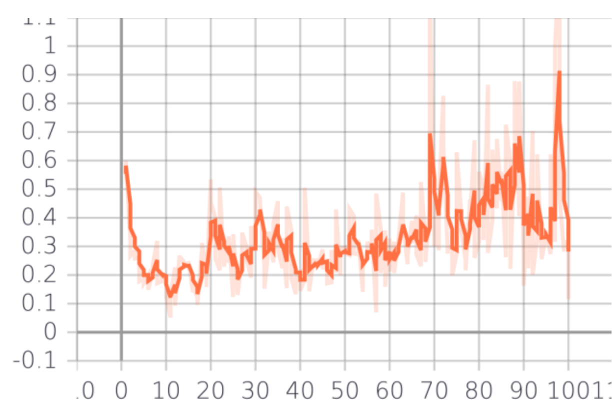

Fig. 4. Alexnet Test Loss

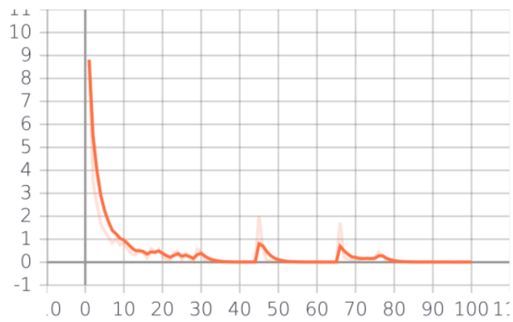
Fig. 5. Lenet Train Loss

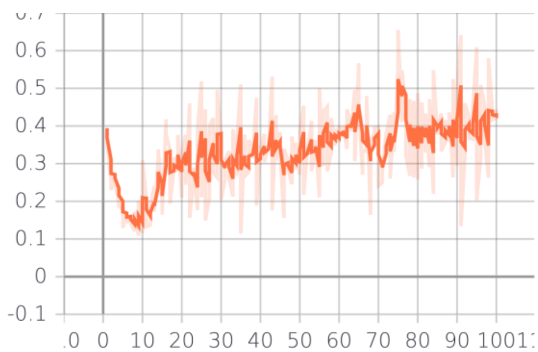
Fig. 6. lenet Test Loss

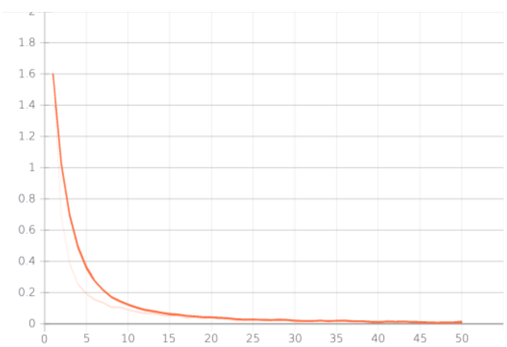
Fig. 7. Wide Residual Networks Train Loss

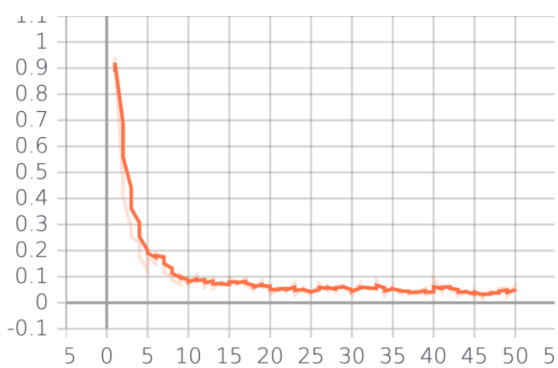
Fig. 8. Wide Residual Networks Test Loss

The above six figures (Fig 3 - 8) are the loss of three neural networks on our training data and validation data. In these figures, the loss of Wide ResNet train and test decreases as epoch increases. However, the loss of the other two neural networks decreases first, and then increases as epoch increases. Besides, accuracy of Wide ResNet on our validation set is 98.58 percentage; meanwhile, accuracies of Alexnet and Lenet neural networks are 96.4 and 92 percentage, respectively. Therefore, we can observe Wide ResNet is more efficient, accurate, and stable on our training and validation dataset. This is the reason that the weighting parameter of Wide ResNet is higher than other two neural networks in boost method.

The other experiment we did was to find dataset for unknown class and to predict them [3]. Intuitively, there were two methods in terms of solving this challenging. First, predict probability of each class for each input. If the input was one of known classes, there would be a probability closed to one which was much higher than all the other classes' probabilities. However, if the input was unknown, there would be a few probabilities of several classes were closed to each other and relatively high. Therefore, a threshold should be set in order to decide whether this input was in one of known classes or not. We initially set the threshold to 0.5. If one of probabilities was greater than 0.5, we decided this input belongs to one of known classes; otherwise, it was not in our known classes. In this way, we tested the probability of belonging to one of known classes for each of 6400 data from EMNIST and MNIST. We observed there were nearly a half probabilities of input data greater than 0.5, and greater threshold did not help. We believed the reason was we did not train other data, so that similarly input data would be sent to one of known classes. For instance, letter 'o' looks like letter 'a', since the probability of 'o' on class 'a' may be greater than 0.5. Thus, this method is not a good idea on determining unknown data.

For the second method, we planned to establish a classifier for other unknown data, and we needed to created a dataset for other class. We only considered letters and numbers since the requirement is to test characters. Therefore, we found letters except 'a' to 'd' and 'h' to 'k', which are 18 letters, and 0 - 9 digits in order to train the model. Since we have 6400 training known data, we planned to find 6400 unknown data in order to keep the balance. Therefore, we added 200 data of each letter and 280 data of each numbers so that there were 6400 unknown data totally, and the number of each unknown letter and digit were also closed to each other. We used EMNIST as our letter database and MNIST as our digit database. Then we appended these unknown data to the original dataset and new hard training dataset became to 12800 data. We changed the order of data to random, and 10-fold cross validation is used to validate the SVM classifier model. We trained SVM model on Intel i7 - 7500U CPU @ 2.7 GHz, 8 GB RAM device. Followings are the result of cross-validation:

TABLE I
10-FOLD CROSS-VALIDATION

No.	Score	Time Consumption (s)
1	0.98515625	72.805
2	0.99140625	73.441
3	0.9875	72.017
4	0.98515625	70.273
5	0.98046875	72.067
6	0.99140625	72.692
7	0.98515625	72.398
8	0.9921875	72.145
9	0.9875	72.039
10	0.98515625	72.737

IV. Conclusions

We have designed and implemented handwritten letter recognition python machine learning and neural network code, and we also analyzed our findings and algorithm performance. We used Lenet, Wide Resnet and Alexnet neural network along with weighting parameter as a boost method to predict both easy and hard dataset. We disrupted the order of the original dataset and using randomly selected validation dataset with equally distribution of each class to test our model. The method we used for out-of-distribution detection was adding additional dataset and SVM classifier with 10 - fold cross validation to filter the known and unknown data as the first step, and then using our neural network module to predict data's label for known data, and predict "other class" directly in SVM classifier for unknown data.

References

[1] Sergey Zagoruyko, Nikos Komodakis: Wide Residual Networks, arXiv:1605.07146 [cs.CV]

[2] Zeyi Wen, Bin Li, Rao Kotagiri, Jian Chen, Yawen Chen, Rui Zhang: Improving Efficiency of SVM k-fold Cross-validation by Alpha Seeding, arXiv:1611.07659

[3] Terrance DeVries, Graham W. Taylor, Learning Confidence for Out-of-Distribution Detection in Neural Networks, arXiv:1802.04865, 2018

[4] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner: Gradient-Based Learning Applied to Document Recognition, Intelligent Signal Processing, 306-351, IEEE Press,2001

[5] Dewi Nasien, Aibollah Haron, Siti Sophiayati Yuhaniz: Support Vector Machine (S VM) for English Handwritten Character Recognition, 10.1109/ICCEA.2010.56, IEEE 2010

[6] Ruyu Zhang; Qingqing Wang; Yue Lu: Combination of ResNet and Center Loss Based Metric Learning for Handwritten Chinese Character Recognr Loss Based Metric Learning for Handwritten Chinese Character Recognition10.1109/ICDAR.2017.324