COMP 551: Mini Project 3- Modified MNIST

GROUP-98: Nikhil Podila, Shantanil Bagchi, and Mehdi Amian

Abstract

In this project, a novel approach is proposed for classification of the Modified MNIST dataset. The task is to identify the numerically largest value among three handwritten digits within each image. The proposed algorithm is a modified version of the well known VGGNet. Since convolutional neural networks (CNN) automatically capture the relevant features, we observe that additional feature selection and preprocessing on dataset are unnecessary. We also observe that data augmentation, optimizer tuning and model ensembling contribute to our best performance. The proposed algorithm reached an accuracy of 99.3% on the test set and ranked the first on Kaggle's Modified MNIST competition.

1 Introduction

In this project, the objective is to identify the numerically largest value digit among each image of a modified version of the MNIST dataset. In recent decade, deep learning techniques show an excellence on machine learning tasks and on different data types e.g. text, speech, image, and video. Based on the lectures of the class, in classical classification algorithms, such as Logistic Regression and Support Vector Machines (SVM), feature selection is an art and plays a key role in the success of algorithm as there is no intrinsic mechanism for feature learning in such models. On the other hand, deep neural networks automatically capture the feature information of the input data. In deep learning, the art is instead, *architecture* engineering.

In the present study, we take advantage of Convolutional Neural Network (CNN) to perform the classification task. We found that a modified version of the well known VGGNet was an ideal architecture for our task. We also found that Data augmentation, hyper parameter tuning and ensemble techniques played a vital role in our test accuracy of 99.3 % on Kaggle's Modified MNIST competition [1].

The next sections are organized as follows: Section 2 addresses the related works in this research area. Section 3 explains the dataset and setup used in the project. Section 4 presents our solution approach using Deep learning. Section 5 describes the results obtained by implementing the proposed approach. Section 6 discusses our inferences from this project and concludes our observations. Section 7 mentions the team's contributions.

2 RELATED WORK

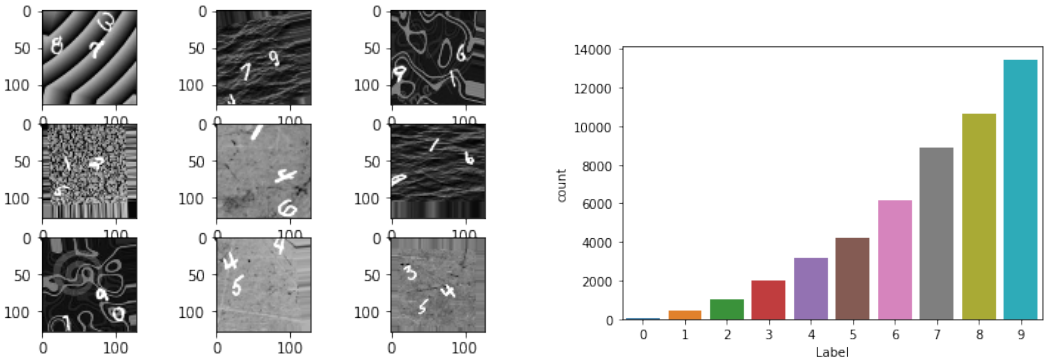
MNIST [2] is a popular dataset in machine learning and particularly in deep learning with 60,000 images of handwritten digits. Image classification using deep learning is a relatively recent advancement. The LeNet architecture was regarded as pioneering architecture of CNN and firstly introduced by LeCun et al. in 1998[3]. Since then a lot of research has been done in this field. An important work is the research conducted in 2017 on Capsnet [4] for classification task. Similarly, in [5] a classifier based on deep residual nets is proposed that obtained the first place in ILSVRC & COCO 2015 competitions. They achieved an error of 3.57% on test set of the ImageNet.

VGGNet [6] was developed by researchers of Google Deepmind and Visual Geometry Group of Oxford University. It was originally created as an up to 19-layer CNN that strictly used 3x3 filters

with stride and pad of 1. Their use of only 3x3 sized filters was quite different from AlexNet's [7] 11x11 filters. The combination of two 3x3 convolution layers had an effective receptive field of 5x5 but decreased the number of parameters and used two ReLU activations instead of one. The number of filters doubled after each maxpool layer. This reinforced the idea of shrinking spatial dimensions, but growing depth. This model got runner up in ILSVRC 2014 with an error rate of 7.3%.

3 DATASET AND SETUP

The training dataset contains 50k training images, and their corresponding labels are present in a separate file of comma-separated value (CSV) format. Each image comprises three MNIST digits rotated and placed on a random pattern background. The size of each image is 128×128 pixels. The test file comprises of 10k images with the same format as the training set. In Fig. 1, some training data samples as well as the histogram of different labels in the entire training data are shown. It can be clearly seen that the distribution of labels is non-uniform leading to an unbalanced dataset.



(a) (b)

Figure 1: Training data: a) Some samples, b) Distribution of the labels

4 PROPOSED APPROACH

4.1 Preprocessing

In a deep learning approach where a large number of parameters exist, a large number of input data samples are needed. It is extremely hard to train a model with small amount of data. Given this fact, an **image augmentation** step has been performed in order to increase the number of inputs and assist the network in solving the problem. In data augmentation, modified copies of images from the dataset are generated. These copies effectively increase the training dataset size and correct for class imbalance in the raw datasets. Modifications are performed by changing the brightness, width, height, applying mirror effect, zooming, and rotation. An example of image augmentation is shown in Fig 2.

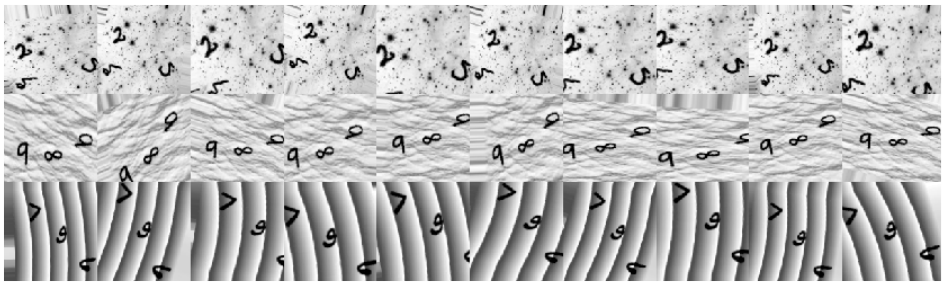


Figure 2: Image augmentation samples: one example in each row

All the images in the given datasets have noise and background textures. We presumed that removing them would aid in training the model. We initially performed **image thresholding** with a threshold of 240 and trained a CNN model. Results using this approach are discussed in the sections 5.1 and 5.2.

4.2 Algorithm Selection

The algorithm used in this project is mostly inspired by the one proposed in [8] which is a modified version of VGGNet. In our algorithm, the pool size of the MaxPooling layers is set to 2×2 instead of 3×3. Also, in original VGGNet, a convolution block consists of three convolutional layer and one MaxPooling layer. In our model, three of such blocks are used. The original batch size of 128 was increased to 200. Dropout layers have been added after each block to combat overfitting.

4.3 Bounding Box Approach

First, we tried the bounding box approach, where the aim is to use Computer vision techniques to create a bounding box around each digit in the image and extract the digits. The bounding box was generated by performing image thresholding and finding contours using procedures specified in article [9]. The digits were then extracted and unskewed using interpolation techniques. The three digits were cropped as separate images each with size of 28×28. The entire processing was carried out using OpenCV [10]. This size is as the same as original MNIST images. Thus, we used a trained (pre-trained) model having high accuracy on MNIST dataset to predict the digits in our processed images. This was followed by manually calculating maximum of the three digits. Results using this approach are discussed in 5.1.

The next approach was stacking the cropped images horizontally and padding the image to a size of 84×84. CNN was used to train on this image data rather than the original MNIST dataset and results are discussed in 5.2.

4.4 Unprocessed training on original dataset

Another method explored was to directly feed in raw images into the CNN models. We presumed that this approach might be useful for this task since theoretically, CNN automatically captures the relevant features from the input data. So, the raw data was passed to the modified VGGNet directly.

5 RESULTS

Several methods with different scenarios have been examined in this project until acceptable results were achieved. The most important results are presented in the following subsections.

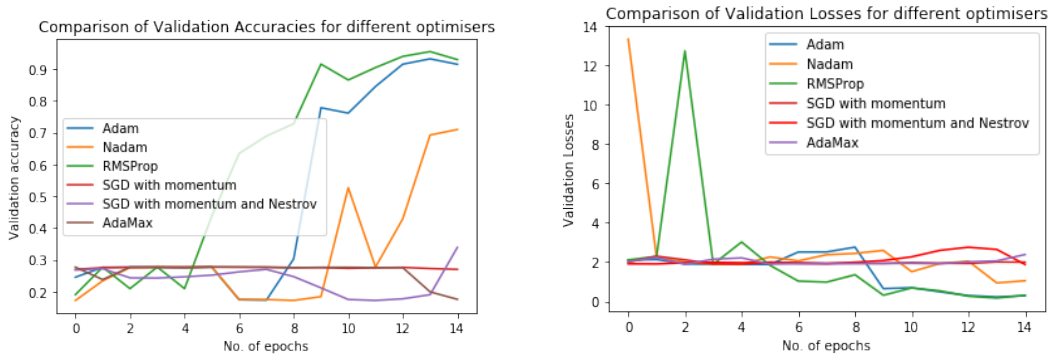
5.1 Bounding Box approach Trained on Original MNIST and Modified MNIST

As a common strategy in image classification, we examined a pretrained model. In other words, we built an appropriate classifier with an acceptable performance for the original MNIST dataset. CNN trained on original MNIST with accuracy of over 99% was used for predicting our processed dataset. Prediction results were poor and could achieve only around 70% accuracy on Modified MNIST training set.

Next, CNN was used to train stacked images of cropped digits from bounding box approach on original dataset. We achieved a maximum accuracy of 95% with this method hinting that a simple approach might work better.

5.2 Modified VGGNet on Modified MNIST Dataset

In this approach, we passed the raw data directly to the CNN model. Example of performance on 15 epochs for different optimizers are shown in Fig 3a and Fig 3b. We selected ADAM optimizer with AmsGrad (True) for our model. An accuracy of around 99% was achieved.



(a) (b)

Figure 3: Validation performance of different optimizers: a) Accuracy b) Loss

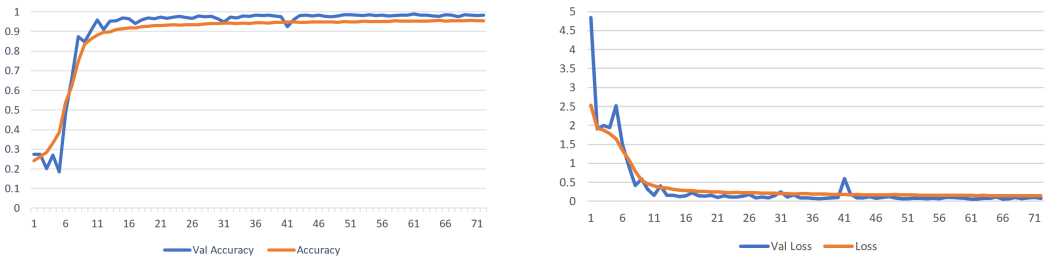
5.3 Hyperparameter Tuning

To determine the best hyper-parameters for data augmentation, grid search was used. We found that rotation angle of 20 worked the best compared to other values. The other parameters are also set smoothly in order to prevent loss of information (e.g. cutoff of letters). The width and height shift ranges are set to 0.1, and the shear and zoom ranges are set to 0.2. The background texture actually provides image augmentation a larger 'inventing space'.

Other tuning trials included changes of dropout value, changing layer variables, trying AMSBound optimizer etc. Learning from the various trials are discussed in the next section.

5.4 Ensemble Approach

One main contribution of this project is taking advantage of the ensemble strategy. Six VGGNets with different random state are stacked. We used a voting method to select the most common prediction results. The ensemble method resulted in an accuracy of 99.3% and ranked us as the first team in the Kaggle's leader board. The performance of this model over different epochs is presented in the Fig 4a and Fig 4b. The confusion matrix for validation set is demonstrated in the Figure 5.

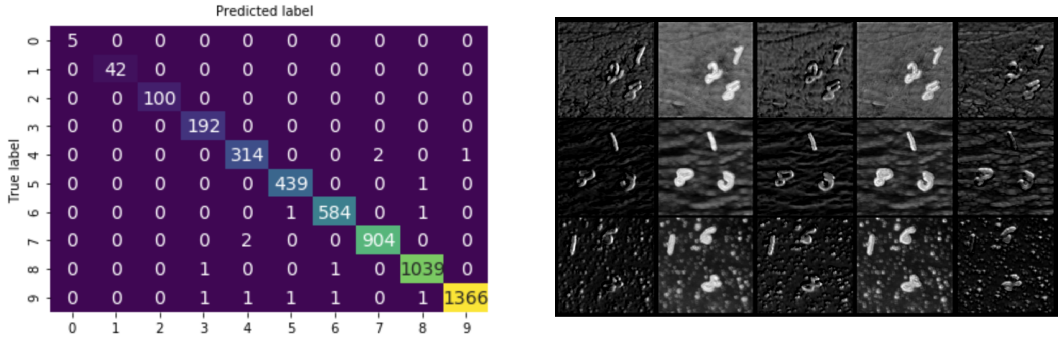


(a) (b)

Figure 4: Performance vs. number of epochs. Blue line: validation, and Orange line: training. a) Accuracy b) Loss

5.5 Feature Maps of CNN models

A major drawback of CNNs is the black-box approach on the end-to-end solution for the given task. Since the traditional procedures of data preprocessing and output transformations are performed within the CNN and not explicitly designed, a series of transformations are being performed by the CNN, which are not visible for validation and ethical analysis.



(a) (b)

Figure 5: a) Confusion matrix b) Feature maps from first layer of CNN model for 3 example images from dataset

Interpretability of deep learning models is essential for certain applications. An example of this is the lack of trust in advanced CNN based tumor detection algorithms among Radiologists, as explained in [12].

In our project, we interpret the models by extracting the feature maps generated from the hidden layer. We then ensure that the algorithm is focusing on the correct areas of the image in order to perform the classification. A few samples from the output of the first layer is shown in Fig 5b. Here, we note that the edge detection is being performed to detect the digits and the background is discarded from most of the feature maps, thus following procedures similar to that used in classical image processing and computer vision.

6 Discussion and Conclusion

When working with data in the format of pictures, each pixel is highly dependent on the neighboring pixels. The most important conclusion of this project is the strength of CNN in handling image data. Despite adding artificial random pattern backgrounds to original hand written image data as well as rotating and placing the digits in random places within each image, the CNN showed an excellent performance in learning complicated properties of the data. Due to natural feature capturing characteristic inherent in deep neural networks, it is instead needed to wisefully design an appropriate architecture and set the hyperparameters. The CNN enables different hidden units in the hidden layers to feed information to each other, i.e. now the network also knows when pixels are close together, or are sitting together in different ways that the network can learn.

One of our other findings is that AmsBound performed similar to Adam with Amsgrad optimizer without much improvement in the validation accuracy as claimed in [11]. The second learning was that dropout layers after each layer helped combat overfitting but changing its value from 0.5 to 0.2 didn't affect the final accuracy though improved the convergence speed. We also learned the importance of interpretability of CNNs to understand their internal approach and correct any errors by visualizing feature maps. Another main contribution of this project was the ensemble approach that resulted in the highest performance and ranked us as the first on Kaggle's leaderboard.

The Modified MNIST dat aset is an ideal basis to train a model for various applications, such as detecting house numbers on street images and vehicle number plate recognition. As a future study, we hope to use this trained model for such applications. In the future, the computation time can be reduced by fitting the dat aset using a smaller architecture, while ensuring the same accuracy. Similarly, exploration of methods to speed up convergence can be explored in detail.

7 Statement of Contributions

The collaborative work of this team involved everyone contributing to the project. Shantanil and Nikhil worked on data preprocessing and analysis, various feature extraction and classifier implementation.

Mehdi worked mainly on data visualisation and hyper parameter tuning. The team collaboratively worked on the report.

References

[1] https://www.kaggle.com/c/modified-mnist

[2] http://yann.lecun.com/exdb/mnist/

[3] Y. LeCun, L. Bottou (1998). Gradientbased Learning Applied to Document Recognition. [online] Available at: http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf

[4] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules, 2017 (https://arxiv.org/pdf/1710.09829.pdf)

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for imagerecognition, 2015.

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

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deepconvolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger,editors,Advances in Neural Information Processing Systems 25, pages 1097–1105. CurranAssociates, Inc., 2012

[8] https://github.com/Bonnie970/Applied_Machine_Learning/tree/master/Assignment3

[9] Satoshi Suzuki and others. Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing, 30(1):32–46, 1985.

[10] Bradski, Gary, and Adrian Kaehler. "OpenCV." Dr. Dobb's journal of software tools 3 (2000).

[11] Liangchen Luo, Yuanhao Xiong, Yan Liu, Xu Sun. "ADAPTIVE GRADIENT METHODS WITH DYNAMIC BOUND OF LEARNING RATE"

[12] https://ai.myesr.org/publications/taking-artificial-intelligence-out-of-the-black-box-an-interpretable-deep-learning-system-for-liver-tumour-diagnosis/