COMP 551 Miniproject 3 Report
Image Classification on Modified MNIST

JianChen Zhao [260787070], Kevin Chen [260658680], and Zhenghua Chen [260783959]

McGill University, Canada

1 abstract

The purpose of this project is to experiment on machine learning models tasked to determine the maximum of 3 handwritten digits on given images. A gray-scale threshold was set to filter out most of the noise in the image, and after randomly applying small affine transformations, the data was given to the models for predictions. Preexisting image classification models and our adaptations were experimented on through this project, since simple models consisting of only a couple convolution layers resulted in very low accuracy. The models in question are VGG16 and ResNet. VGG16 ended up with more than 98 percent accuracy while ResNet only achieved 97 accuracy.

2 Introduction

Computer vision has become a popular topic in various field. Since the machine learning and computer performance break through, computer vision is able to understand the information of the images. However, most of the images do not have identified format or style, which increases the difficulty of extracting features from the image. MNIST is one of the most common datasets used in computer vision for teaching. With appropriate preprocessing, MNIST can be predicted with many kinds of model including non neural network models. The dataset given for this project is a more complex alteration on the original MNIST. Each image in this modified dataset includes three black handwritten digits and background noise. The digits may or may not be repeated in a single image. Our models aim to find the maximum of the digits in these images. To reduce training time, the computation is performed on GPU instead of CPU, by using high level library such as TensorFlow and PyTorch. A small CNN is used as a baseline model, while VGG16 and ResNet50v2 are used to achieve high accuracy predictions.

3 Related work

Convolutional neural networks are widely used to classify images by using convolution filters to learn and extract patterns. The VGG and ResNet models are both fairly simple models that achieved good results on ImageNet.

VGG uses small three by three filters, resulting in fewer parameters to train, faster convergence, reduced overfitting. [4]

The ResNet models improve on this by using residual blocks to mitigate vanishing gradients, thus allowing more layers in a model. [1][2]

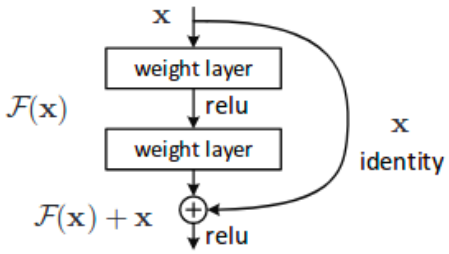

Figure 1: ResNet's vanishing gradient solution. [1]

4 Dataset and setup

The project dataset contains 50,000 gray-scale images with dimension of 128 x 128. Each image contains 3 handwritten digits in black and a background noise.

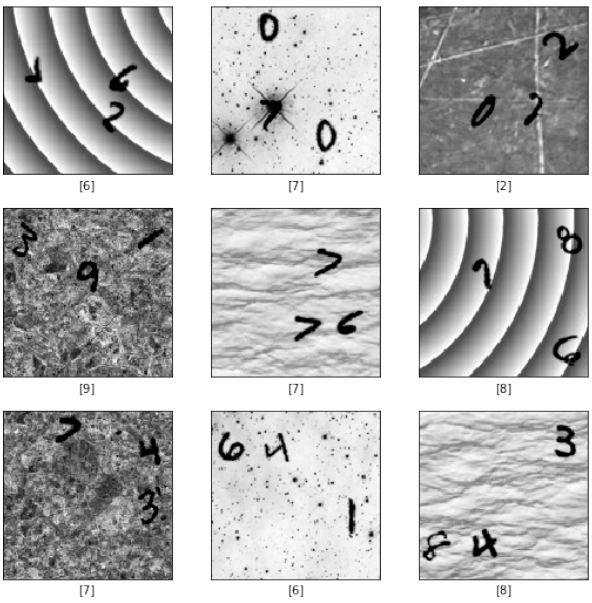

Figure 2: Examples of images taken from the dataset.

Grey-scale values are represented by unsigned 8 bit integers. To simplify the computation, we've converted them to floating point by dividing by 255. Black is represented by 1, and white is represented by 0. As observable in figure 2 and 3, the digits have gray-scale value very close to black. The noises on the other hand, are mostly grey, and have values close to 1.

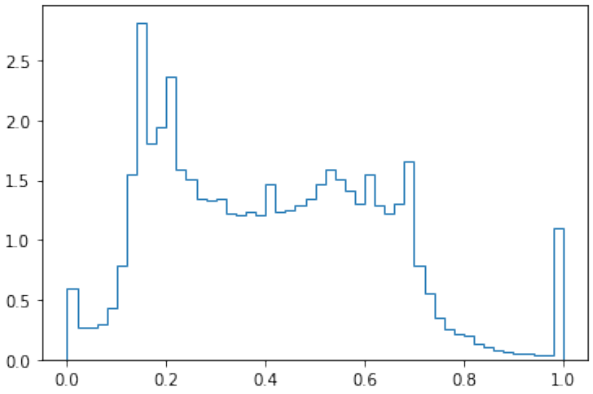

Figure 3: Histogram of pixel values showing the range of noise.

Thus to potentially improve the accuracy of the model, we've filtered the noise by simply clamping the values between 0.8 and 1, then scaled back to between 0 and 1. The resulting images in figure 4 appears easier to classify, even to the human eye.

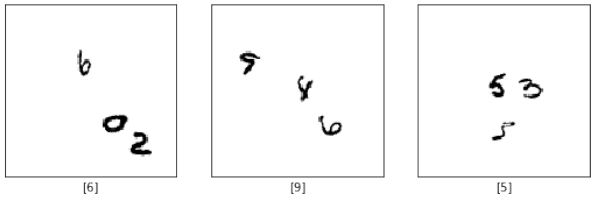

Figure 4: Examples of filtered images taken from the dataset

The dataset have 10 different classes, corresponding to the maximum digit of the images. However, these classes are heavily imbalanced as shown in figure 5. The is maybe due to the fact that the digits in each image are likely chosen uniformly, thus the probability of an image with maximum 0, which requires three 0's appearing, is significantly lower than that of an image with higher maximum.

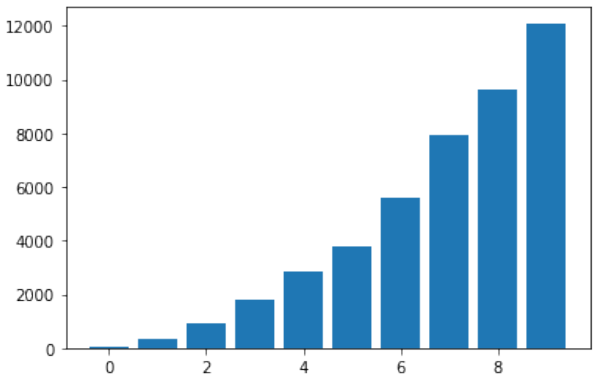

Figure 5: Distribution of the classes.

To deal with the imbalance, we used data augmentation and resampling. To reduce overfitting, we augmented the data by adding small random affine transformations to individual images. Because of the small transformations, the images needed to be resized so that the digits don't get cropped. We also resampled the data such that the resulting distribution is uniform.

5 Proposed approach

In our experiments, we compare the accuracy of 3 different model.

The first model we used is a simple model consisting of 3 convolution layer(conv3), based on architectures described by Simonyan and Zisserman [4]. Using filters of size 3 by 3, we hope that the layers will learn pattern that corresponds to the digits.

The second model we experimented on is VGG16 [4]. It consists of 5 blocks of conv3 layers with max pooling after every blocks. By comparing this model to our initial model, we can directly see the effect of adding more layers. However, adding too many layers might not always improve the accuracy. For example, VGG19 performs worst in ImageNet than VGG16 [4]. This could be caused by vanishing gradients, where the first layers' error gradient are so low that they don't get updated correctly.

The third model, ResNet50v2 [1][2], solves this problem and allows for more layers by using residual blocks. It works by having "skip layers" (figure 1) to allow the error gradient to backpropagate properly. This model also uses batch normalization, which helps with training time, which is essential since the model has a lot more layers.

To adapt VGG16 and ResNet50v2 to our problem, we first need to increase the dimensionality of our images to have three channels. We do this by adding a convolution layer of 3 filters with size 1 by 1 (this method is also used on the simple method for consistency). Then, to reduce overfitting on all three models, we add a dropout layer before the top layer. The models are trained with rectified Adam optimizer [3] with categorical crossentropy.

The models are trained using a rectified Adam optimizer [3]. It warms up the learning rate and applies decay as the training runs. We also use early stopping when the validation accuracy stabilises during the training of the models to prevent overfitting and consistently stop the training for comparison.

	simple	VGG16	ResNet50v2
convolution	3	13	49
dense	3	3	1

Table 1: Number of layers of the different models used.

Table 1 shows a comparisons of the number of layers of the models. The count doesn't account for the first convolution layer used to increase the number of channels to 3. Note that ResNet has only one dense layer, used for prediction.

6 Results

6.1 Simple Model

	training	validation
original	0.2685	0.2722
denoised	0.9760	0.9356
augmented	0.9590	0.9582
resampled	0.9766	0.9754

Table 2: Training and validation accuracy of simple model on original, denoised, augmented and then resampled dataset.

Our simple model did not perform well with the unprocessed data. It has a validation accuracy of only 0.2722. This number corresponds roughly to the percentage of samples in class 9, so our model didn't learn. With the noise removed, the model was able to learn and achieve a 0.9356 validation accuracy.

However, we notice that the model is overfitted as the training accuracy is higher than the validation accuracy. With augmented data, the training accuracy dropped because of the added complexity of random transformations on the images. Instead, the validation accuracy increased and got very close to training accuracy, showing that it helped reduce overfitting.

With resampled data, we could achieve even higher accuracy, by correctly classifying smaller classes more frequently. The overfitting is almost unnoticeable, suggesting that the data augmentation works well, specially when some data are oversampled.

6.2 VGG16

	training	validation
original	0.9828	0.9770
denoised	0.9872	0.9748
augmented	0.9729	0.9714
resampled	0.9800	0.9806

Table 3: Training and validation accuracy of VGG16 on original, denoised, augmented and then resampled dataset.

VGG16 performed better overall. We can see that denoising the images had no effect, withing some variance, on the accuracy. This shows that larger model can handle more complex data, and achieve high accuracy even compared to models trained with preprocessed data. Because of the extra convolution layers, VGG16 was able to learn deeper patterns. This results in less overfitting, as the model is able to recognize more and smaller patterns.

Augmenting the data doesn't seem to improve the accuracy either. Although it helped reduce overfitting, the overall accuracy is lower compared to unaugmented data, caused by the increased complexity and reduced overfitting.

Similar to the simple model, with resampling of the data, the model was able to perform better.

6.3 ResNet50v2

	training	validation
original	0.9597	0.9518
denoised	0.9785	0.9556
augmented	0.9755	0.9751
resampled	0.9799	0.9782

Table 4: Training and validation accuracy of ResNet50v2 on original, denoised, augmented and then resampled dataset.

With unpreprocessed data, ResNet performed worst than VGG16. One possibility is that the top most

convolution layers are worst at approximating the maximum function than dense layers. Another reason is that, because the model is more complex, and images are simple, especially the denoised data, overfitting happens faster.

Consistently with previous models, augmenting the data mitigated some overfitting and accuracy increased, and resampling the data increased it even more.

7 Discussion and Conclusion

Three different models were trained on a modified MNIST dataset. The simple models shows that the images are fairly easy to classify accurately with small models, more so when it is preprocessed correctly. As such, models that have a lot more trainable parameters, such as ResNet50v2, overfitting becomes a problem. Techniques such as data augmentation and resampling of the imbalanced classes can improve accuracy by reducing overfitting. For simple image classification tasks, small models such as VGG16 is accurate enough and has the benefit of fast training time. In fact, we achieved our highscore of 0.98033 on Kaggle with VGG16.

Although we've only applied simple preprocessing steps, there are more rigourous methods, such as ZCA whitening that decorrelates the image pixels. This seems a viable solution since the noise are patterns and could potentially be filtered out more efficiently.

8 Statement of Contributions

- Kevin Chen: CNN models and test with different architectures (did not end up in report because of complexity and low accuracy), and writeup;
- JianChen Zhao: Dataset visualization, preprocessing and augmentation, CNN, VGG and ResNet models testing, and writeup;
- Zhenghua Chen: VGG and ResNet model testing and tuning, and writeup.

References

[1] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[2] Kaiming He et al. *Identity Mappings in Deep Residual Networks*. 2016. arXiv: 1603.05027 [cs.CV].

[3] Liyuan Liu et al. *On the Variance of the Adaptive Learning Rate and Beyond*. 2019. arXiv: 1908.03265 [cs.LG].

[4] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. arXiv: 1409.1556 [cs.CV].