COMP 551 - Winter 2019
Mini-Project 3

Victoria Chatron-Michaud - 260710584
Nick Chahley - 260865097
Brince Jones - 260862794

McGill University, Montréal, Québec, Canada

Abstract

For this project we were tasked with performing image analysis on a variant of the Modified National Institute of Standards and Technology (MNIST) dataset. The objective of this project was to develop a machine learning model that correctly identifies the hand written digit in each image which covers the most space. To process the images a binary threshold was applied to remove background noise. Then the largest digit in the image was identified. Finally, all other digits in the image were removed to prepare for classification. Due to the exceptional performance on the *ImageNet* classification dataset, the Residual Neural Network (ResNet) model was selected to classify the processed images. Additionally, a Support Vector Machine (SVM) model was trained as a baseline to compare ”MnistResNet” with the performance of a more traditional machine learning methodology. ResNet18 (top F1=0.910) greatly outperformed the SVM (F1=0.207). Interestingly, pre-processing methods seemed to have a negative impact on the accuracy of the neural network.

Keywords: Image Analysis, Computer Vision, Machine Learning, ResNet, Neural Network, Support Vector Machine

1. Introduction

For the project we were tasked with performing image analysis on a variant of the Modified National Institute of Standards and Technology (MNIST) dataset.The original MNIST database is comprised of handwritten digits separated into a training and test set. The training set is composed of hand written digits from employees of the American Census Bureau, while the test set is made from digits written by American high school students. This dataset is commonly used to test machine learning models because of the wide variation and non-uniformity in the digits from two separate groups. The images used to train and test our model include multiple digits from the MNIST dataset that differ in size and orientation. These digits are also placed over a background of random noise. The images use a single grey-scale channel of values rather than a multi-channel color image. The objective of this project was to develop a machine learning model that correctly identifies the digit in each image which covers the most space.

2. Related Work

The MNIST dataset has been used as a benchmark for machine learning models for decades [5]. LeCun et al. 1998 use this dataset in their seminal paper, *Gradient-based learning applied to document recognition*, as way to demonstrate gradient-based learning and Graph Transformer Networks. They determined that Convolutional Neural Networks outperformed other models at recognizing hand written digits [5].

More recently, Simard et al. 2003 used the MNIST dataset to compare and contrast several different neural network methodologies. Similar to LeCun et al.1998, they found that simple Convolutional Neural Networks that did not leverage more advanced techniques such as momentum, averaging layers, or weight decay performed better than fully connected neural network models. This paper also emphasizes the importance the size of the training set has one the performance of these types of models. Simard et al. 2003 found that increasing the size of the training set was the most effective way to increase performance [7].

State of the art methods for attacking a problem such as identifying MNIST consist of committees of these neural networks. These approaches can achieve human levels of accuracy on the MNIST classification problem. Cireşan et al. 2012 describe one such approach using a committee of 35 Convolutional Neural Networks [1].

Currently one of the best performing models for image classification is the Residual Convolutional Neural Network or ”ResNet”. This model won the 2015 International Conference on Computer Vision competition, recording a test set error improvement of 26% from the wining model the previous year. ResNet’s test set error on the *ImageNet* classification dataset was 4.94%, making it the first model to surpass human performance in image classification on this dataset [4]. Since the ResNet model described excels at image classification tasks, it was chosen as our model for classifying the MNIST digits. Additionally, a Support Vector Machine (SVM) model was trained as a baseline to compare ”ResNet” with the performance of a more traditional machine learning methodology. The Support-Vector Network [3] creates a linear decision space, or hyperplane, to separate training data into categories, even when data is not otherwise lineary separable.

3. Data-set and Setup

The MNIST dataset provided was in the form of 40000 64x64 grayscale images, each with an associated label indicating the largest handwritten number contained in it. Each image contains one or multiple numbers.

3.1. Preprocessing

Several methods to preprocess the MNIST images and single out the largest handwritten number are available via OpenCV[2], a library of programming functions created to facilitate computer vision practices. The steps to process the images are as follows:

1. Binary Thresholding : To remove background noise, leaving only the observed digits in the image. This is performed for all sorting methods.
2. Find the largest number: Using one of three methods, sort the contours of the numbers from the thresholded image.
3. Remove all other digits: Mask over all other digits leaving only the largest in the final processed image.

The sorting methods to find the largest number are as follows:

- Largest rotated rectangle
- Largest rotated ellipse
- Greatest number of pixels

3.1.1. Largest Rotated Rectangle

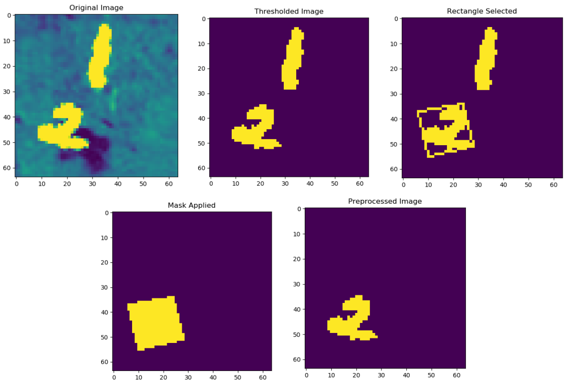


Figure 1: Largest Rotated Rectangle Steps

Figure 1 shows the first preprocessing method, in which we find the largest number (contour) via sorting the areas of the contours from their minimum bounding rectangles. The area of the rectangle is found by multiplying its height and width. The digit with the largest minimum bounding rectangle was passed to the processed dataset.

3.1.2. Largest Rotated Ellipse

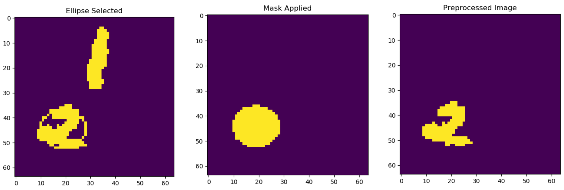


Figure 2: Largest Ellipse Steps

Figure 2 shows how we find the largest digit by sorting the areas of the minimum bounding ellipses from each contour. The area of the ellipse is found by multiplying its major and minor axes by pi. The digit with the largest minimum bounding ellipse was passed to the processed dataset.

3.1.3. Greatest Number of Pixels

The third method to find the greatest number of pixels was found by getting the area of each contour from the thresholded image, and keeping the digit with the largest area to pass to the dataset.

4. Proposed Approach

4.1. SVM

Linear SVC is performed to provide a simple linear classifier, maximizing the margin around class-separating hyperplanes of the given data[8]. Linear SVC is a form of SVC (SVM) in which the we have a linear kernel for the basis function, which is chosen over simple SVC due to its lower bias/higher variance tradeoff. Furthermore, Linear SVC performs better with large amounts of data[6]. This method is expected to perform at a lower accuracy due to its simplicity and due to the MNIST dataset having 10 classes (digits 0-9). We implemented our SVM using sklearn. A bias term is added to the numpy array of images so that the learned hyperplane does not pass through the origin, providing a better performing model.

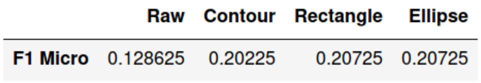
4.2. ResNet

ResNet was originally developed for competing in *ImageNet*, a color (tri-channel) image classification task with 1000 classes. We adapted the network for use with our MNIST-style data by creating a PyTorch ResNet object and redefining the input layer (self.conv1) to accept single-channel images, setting the number of classes to 10, and adding a softmax function to the end of the forward pass. We defined and tested "MnistResNet" models of different depths adapted from ResNet 18, 50, and 152. The models were trained on a 80/20 training/validation split of the labeled images with a training batch size of 64 and a validation batch size of 256.

5. Results

5.1. SVM

Figure 3 shows that the F1 score of the SVM increased with image pre-processing. The largest rotated rectangle and largest rotated ellipse (F1 = 0.207) outperformed classification using raw image data (F1 = 0.129) and using contours only (F1 = 0.202).



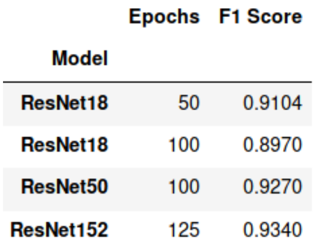
	Raw	Contour	Rectangle	Ellipse
F1 Micro	0.128625	0.20225	0.20725	0.20725

Figure 3: Validation F1 scores of linear SVM models trained on the raw dataset and each of our preprocessing sorting methods. The number of epochs for each model represents a stopping point where we observed the best F1 performance. ResNet18 trained to 100 epochs is an exception, and demonstrates overfitting occured when the model is trained far past an optimal epoch range.

5.2. ResNet

Overall there was a trend for the deeper networks to outperform their shallower variants, with ResNet152 (F1 = 0.934) outperforming ResNet50 (F1 = 0.927) and ResNet18 (top F1 = 0.910). We found that the number of epochs before a model's training curve plateaued or the model began to overfit the training data increased with network depth. This is demonstrated in Figure 4, where ResNet18 plateaued at 50 epochs and began to overfit by 100 epochs, while ResNet50 and 152 maximized performance at 100 and 125 epochs, respectively.

Image preprocessing did not improve the performance of our top ResNet. Figure 5 shows that our ResNet152 network displayed lower accuracy during validation and higher loss during training when given a preprocessed dataset. This draws us to further questions on preprocessing methods (see Section 6).



	Epochs	F1 Score
Model		
ResNet18	50	0.9104
ResNet18	100	0.8970
ResNet50	100	0.9270
ResNet152	125	0.9340

Figure 4: Validation F1 scores for MnistResNet models trained on raw images to a range of epochs.

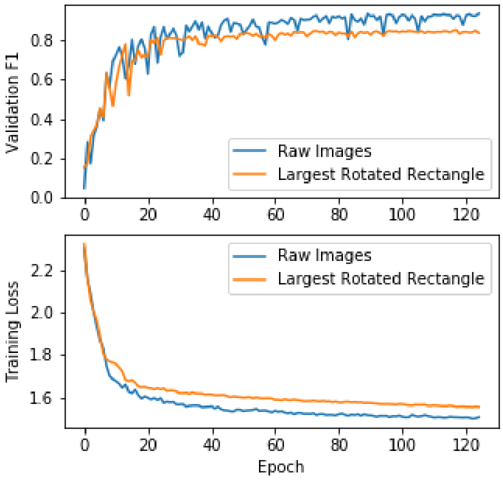


Figure 5: Training curves of MnistResNet152 with raw and preprocessed (largest rotated rectangle) images.

6. Discussion and Conclusion

As expected, the ResNet model greatly outperformed Linear SVC, with ResNet152 showing a 450% F1 score improvement compared to the SVM using the largest rotating polygon method. Interestingly, preprocessing methods seemed to have a negative impact on the accuracy of the neural network. This could potentially be due to the reductive nature of processing. Preprocessing data typically serves to ensure the data is useable by the machine, however, this also acts to simplify the data. In the case of a neural networks that excel in identifying patterns from very detailed and complex data, simplifying the images with preprocessing might be removing components of the data that limit the neural network.

We would like to further explore preprocessing tweaks such as the values used in binary threshold, due to our results and the lack of improvement from preprocessing. As it is possible that the threshold is discarding too much of the digits, time permitting, we would like to explore different grayscale threshold values and their effects on performance.

7. Statement of Contributions

- Victoria Chatron-Michaud was responsible for image preprocessing, Linear SVC implementation, and report writing.
- Nick Chahley worked on ResNet implementation and running models.
- Brince Jones contributed to the written report.

8. References

[1] Ciresan, Dan, Ueli Meier, and Jrgen Schmidhuber. *Multi-column deep neural networks for image classification*. arXiv:1202.2745. 2012.

[2] Contour Features. *OpenCV*, Doxygen, 18 Dec. 2015, docs.opencv.org/3.1.0/dd/d49/tutorial_py_contour_features.html.

[3] Cortes, Corinna, and Vladimir Vapnik. Support-Vector Networks. Machine Learning, vol. 20, no. 3, pp. 273297., doi:10.1007/bf00994018. 1995.

[4] He K, Zhang X, Ren S, Sun J. *Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification*. Proceedings of the IEEE international conference on computer vision, 1026-1034, 2015.

[5] LeCun Y, Bottou L, Bengio Y, Haffner P. *Gradient-based learning applied to document recognition*. Proceedings of the IEEE, 86(11), 2278-2324, 1998.

[6] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[7] Simard PY, Steinkraus D, Platt JC. *Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis*. ICDAR,958, 2003.

[8] Sklearn.svm.LinearSVC. Scikit Learn, 2011, scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html.