COMP-598 Mini-Project #3
Handwritten Digits Classification

David Rapoport, Zafarali Ahmed, Pascale Gourdeau
Team Name: TBD

I. Introduction

The MNIST database**??** of handwritten digits is often used as a baseline to compare performance of different classifiers. In this paper, we use a modified version of the MNIST database where the digits have been modified by (1) embossing, (2) rotation, (3) rescaling, and (4) random texturing. This makes the problem much more difficult and we demonstrate poor performance with linear classifiers.

In this work we explore the performance of three classifiers: (1) Logistic Regression, (2) Support Vector Machine and (3) The Feed-forward Neural Network.

¡Something about results¿

We then explore two algorithms, the tried and tested Convolution Neural Network **????** and the brand new Spatial Transformer Network **??** to find that ......

II. Related Work

III. Data

The dataset was obtained from the Kaggle Competition Website. It is a modification of the MNIST **??** database. A sampling of some digits can be seen in Fig **??** and it is apparent that this proves to be a difficult task for even humans to distinguish. In particular, we expect the digits 6 and 9 to be almost indistinguishable.

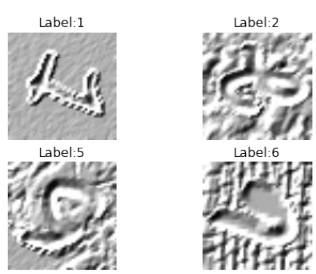

Fig. 1. A sampling of the modified MNIST database

IV. Methodology

We used the NumPy package and the scikit-learn library to perform feature extraction and selection, implement our classifiers, and analyse our results.

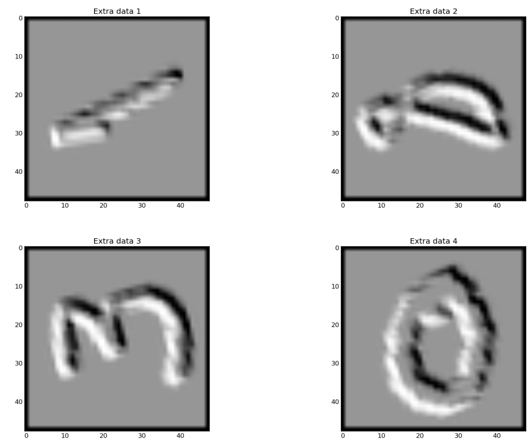

Fig. 2. Four example image from the ”Extra Digits” dataset

A. Feature Extraction and Preprocessing

B. Feature Selection

C. Generating Extra Data

As per the suggestions of Patrice Simard et al. [6] we decided to augment our learners with extra data. The data generated came from two different sources and the effect of the addition of either supplementary dataset is explored later in this paper.

The first additional datasource involved transformations of the original MNIST dataset [2]. We wil refer to this dataset as ”Extra Digits” throughout the rest of the paper. First we enlarged the images from 28×28 to 48×48. Then we rotated by a random angle $\theta \in [0, 359]$. Finally we applied an emboss to the image. We attempted to add a pattern to the new image as in the modified MNIST dataset, but were unable to find a pattern which we felt maintained the integrity of the image in a way which the patterns used in the modified MNIST dataset did. An example of images from the dataset is shown in **??**

The second datasource we attempted to generate was motivated by Patrice Simard et al. [6]. This involved performing an affine deformation of the modified MNIST image. We refer to this additional dataset as ”Perturbed Modified Digits” throughout the rest of the paper. The methodology for generating these perturbed digits is as follows. Given a $n \times n$ image, we first generate two random $n \times n$ displacement fields $?x(x', y') \rightarrow uniform(-1, 1)$ and $?y(x', y') \rightarrow uniform(-1, 1)$. We the convolve these displacement fields by a gaussian filter with $\mu = 0$ and σ being a variable to the function which defaults to 3. After this we normalize the displacement fields by dividing

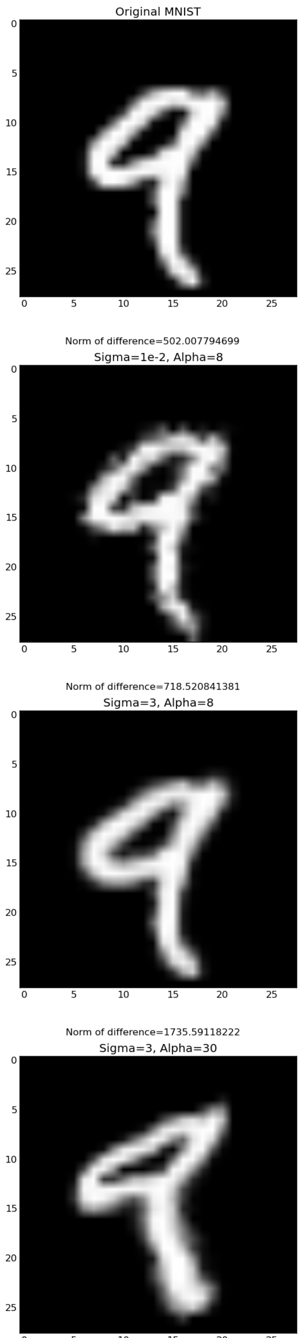

Fig. 3. The effect of varying σ and α

each element in the field by the norm of the matrix. All the elements in both displacement fields are then multiplied by α which is a variable to the function. We then generate a new $n \times n$ image as such: for each pixel (i, j) in the original, the new value at that pixel is the value at $?x(i, j), ?y(i, j)$ in the original picture. Bilinear interpolation is used to determine the value of $new(i, j) = original(?x(i, j), ?y(i, j))$ when $?x(i, j), ?y(i, j)$ is not an integer. For indices which appear out of bounds of the picture we simply used a default value of 0. **??** shows the effect of varying α and σ. The norm of the difference between the original MNIST image and the modified image are included in the pictures.

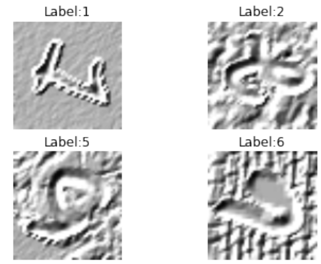

Fig. 4. A sampling of the modified MNIST database

D. Classification Algorithms

1) Baseline: Logistic Regression: In Logistic Regression, we want to estimate the probability that some random vector $X = (x_1, \ldots, x_n)$ has a class $Y = y_k$, $P(Y = y_k | X = (x_1, \ldots, x_n))$. In the binary case we can derive the following using Bayes rule and conditioning:

$$P(Y = 1 \mid X) = \frac{P(X, Y = 1)}{P(X)}$$
$$= \frac{P(X \mid Y = 1) \cdot P(Y = 1)}{P(X \mid Y = 1) \cdot P(Y = 1) + P(X \mid Y = 0) \cdot P(Y = 0)}$$
$$= \frac{1}{1 + \exp(-a)} = \sigma(a) \text{ (Sigmoid Function)}$$

where $a = \ln\left(\frac{P(Y=1 \mid X)}{P(Y=0 \mid X)}\right)$ is the log-odds ratio. By approximating the log-odds ratio as a linear decision boundary of the features and weights, $w^T x$ we can use this as an estimate of the class being $Y = 1$. We can optimize the Log-likelihood or Cross-Entropy function:

$$L(w) = -\sum_{i=1}^{n} y_i \log\Big(\sigma(w^T x_i)\Big) + (1 - y_i) \log\Big(1 - \sigma(w^T x_i)\Big) \tag{1}$$

and search for the optimal set of weights using the *gradient descent algorithm* with N steps and update rule **??**:

$$w_{k+1} = w_k + \alpha_k \sum_{i=1}^{n} \Big(x_i\big(y_i - \sigma(w_k^T x_i)\big)\Big) \tag{2}$$

2) SVM:

3) Fully Connected Feedforward Neural Network:

4) Convolution Neural Network: The convolution neural network **??** is a neural network with specialized layers in which not all neurons are connected to each other. Infact the main component is the subblayer (Fig **??**) which contains the Convolution2D and MaxPool2d pair of layers. A convolution of an image is the result where each pixel is the weighted sum of its neighbouring pixels (moving window weighted sum). This means that our 2D convolution takes a weighted sum of each neighbouring pixel of the handwritten digit. *num_filters* is the number of moving windows of size *filter_size*

$\times$ *filter_size* we do convolutions with. In MaxPooling we select the maximum in a region of size *pool_size* $\times$ *pool_size* pixels obtained from the 2D convolution.

As shown in the (simplified) figure, the convolution conducted on the input is transformed into the 3D spatial arrangement of the filters. This is then subsampled (by selecting the MAX element from each subsection of the filter) to produce the set of outputs that can be reused in future layers. In our architecture we stacked K of the Convolution2D-Maxpool sublayers ($K = \{1, 2, 3, 4\}$) and finally ran the outputs through a fully connected dense layer which had the same number of units as *num_filters* in the last Convolution2D-Maxpool subplayer. To obtain the final output, we ran this through a $p = 0.5$ dropout and then into a fully connected dense layer of size 10 to make the predictions. This is summarized in Fig **??**.

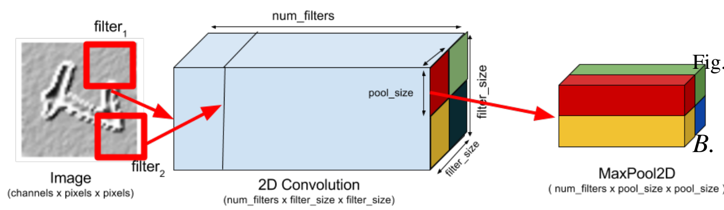

Fig. 5. A simplified example of the sublayer containing Convolution2D layer and a MaxPool2D layer. The variables correspond to the authors implementation of the network. The final network used one,two and three of these sublayers in tandem.

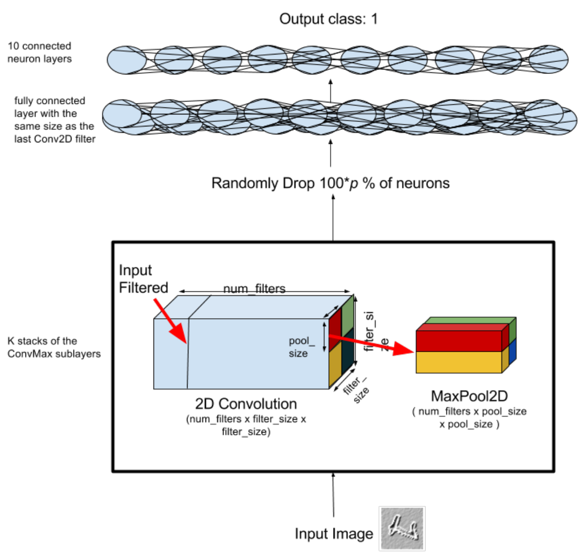

Fig. 6. The Convolution Neural Architechture included k stacked sublayers shown in Fig **??** connected to a dropout layer and finally fed into a fully connected layer containing 10 neurons that did the final classification.

5) Spatial Transformer Network:

E. Cross-Validation and Choice of Hyperparameters

V. Results

A. Convolution Neural Network

ROC (?) curves are shown in

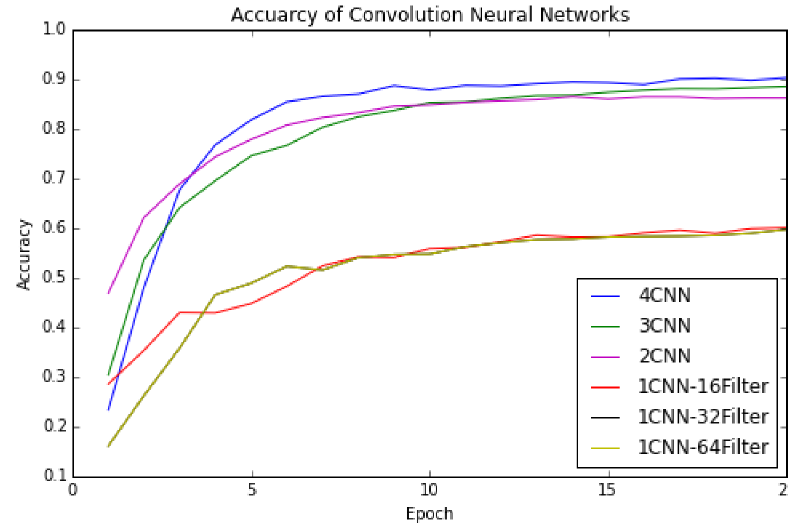

Fig. 7. ROC Curves for our CNN architectures

B. Spatial Transformer Network

VI. Discussion

A. Feature Extraction and Selection

B. Classifier Performance

C. Future Work

VII. Statement of Contributions

VIII. Integrity of work

We hereby state that all the work presented in this report is that of the authors.

References

[1] Lecun Y., Bengio Y., and Haffner P. *Gradient-Based Learning Applied to Document Recognition* Proceedings of the IEEE, 1998.

[2] Lecun Y., Cortes C. Burges C. *The MNIST database of handwritten digits*

[3] Sebastini, F. *Machine Learning in Automated Text Categorization* ACM Computing Surveys (CSUR) 34 2002.

[4] Claudiu D. and Meier U. et al. *Convolutional Neural Network Committees For Handwritten Character Classification* ACM Computing Surveys (CSUR) 34 2002.

[5] Jaderberg, M., et al. *Spatial Transformer Networks* ArXiv 2015

[6] Simard F., Steinkraus D., Platt J. *Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis* Microsoft http://research.microsoft.com/pubs/68920/icdar03.pdf.

[7] LeCun Y., Cortes, C. and Burges C.J.C. *The MNIST Database of handwritten digits* http://yann.lecun.com/exdb/mnist/