# Handwritten Digit Classification

Ian Forbes, Xinchi Chen, Niladri Mohanty

November 3, 2014

## 1 abstract

We attempt to classify images of handwritten digits written on various textured backgrounds. Using a training set of 50000 images we evaluate the performance of several popular machine learning algorithms.

## 2 Introduction

## 3 Preprocessing

The most basic method to reduce noise from any signal is to average it across lot of samples. We tried to average all images for the same class to get rid of noise. As the noise (texture) is random, it is not a great technique to reduce the noise.

We tried many different preprocessing methods in attempt to reduce images noise, sharpen the appearance of the target digit, and most importantly improve our algorithms' accuracy. The first and most basic of theses methods was to normalize all of the images to a $[0, 1]$ scale. The reason for this decision was that while browsing through the images we noticed that many were mostly gray with very little black or white. We believed that by normalizing the images there would be more contrast between the digit and the background which would make it easier to identify the digit.

While normalization proved promising in validation, increasing our validation accuracy by roughly 1%, our actual Kaggle submissions scored lower than the un-normalized submission.

Another method we tried was image sharpening. Image sharpening works by subtracting a blurred copy of the image from a scaled version of the original image. In particular we tried subtracting

## 4 Feature Selection

[?] Blockwise histograms of local orientations is used to recognize objects. This is called as Pyramid Histogram of Oriented Gradients (PHOG). Each pixel in the image is assigned an orientation and magnitude based on the local gradient and histograms are constructed by aggregating the pixel responses within cells of various sizes. The input grayscale image is convolved with filters which respond to horizontal and vertical gradients from which the magnitude and orientation is computed. The orientation could be signed (0360) or unsigned (0180). The signed gradient distinguishes between black to white and white to black transitions which might be useful for digits. PHOG features are tested with Support Vector Machine (SVM) Gaussian kernel. The SVM gives 42.06

We also tried to remove noise by utilizing Fourier Transforms. FFT can help to provide new ways to do familiar processing such as enhancing brightness and contrast, blurring, sharpening and noise removal. Fourier Transforms is the key to remove noise (small features) from an image, while preserving the overall larger aspects of the image. Although after inverse FFT, images are more clearer than the raw images and digits are recognisable. But again SVM gives 11.33

# 5 Algorithms

## 5.1 Naive Bayes

Naive Bayes is a basic machine learning algorithm based on Bayes rule that can be used for multiclass classification. It uses Bayes rule along with the assumption that all features are conditionally independent given the class. The decision rule for Bayes rule is given by:

$$\operatorname*{argmax}_c \Pr(C = c) \prod_{i=0}^{n} \Pr(F_i = f_i | C = c)$$

We used a variant of Naive Bayes called Multinomial Naive Bayes. This method creates a Multinomial probability distribution for each feature given the class. In technical terms we estimate the distribution $\Pr(F_i = f_i | C = c)$ over each features $f \in F$ and each class $c \in C$ from our training data. Then we use the decision rule to find the class $c$ which maximizes this probability when classifying a test example.

## 5.2 Neural Network

## 5.3 Support Vector Machine

Support vector machines (SVMs) have been a promising tool for data classification. Its basic idea is to map data into a high dimensional space and find a separating hyperplane with the maximal margin. We already discussed different feature selection strategies. [?] Combination of features and SVM is used to classify using SVM classifier. In all cases, model selection was performed using a validation set. SVM with Gaussian kernel is used. The Guassian kernel function is given by following with  ranging from 10-7 to 1 and C = 0.1 to 105.

Our target was to achieve best validation and test accuracy for

[?]

## 5.4 Convolutional Neural Network

# 6 Discussion

# 7 References

@MISCMaji09fastand, author = Subhransu Maji and Jitendra Malik, title = Fast and Accurate Digit Classification, year = 2009 @inproceedingslarochelle2007empirical, title=An empirical evaluation of deep architectures on problems with many factors of variation, author=Larochelle, Hugo and Erhan, Dumitru and Courville, Aaron and Bergstra, James and Bengio, Yoshua, booktitle=Proceedings of the 24th international conference on Machine learning, pages=473–480, year=2007, organization=ACM @incollectionchen2006combining, title=Combining SVMs with various feature selection strategies, author=Chen, Yi-Wei and Lin, Chih-Jen, booktitle=Feature extraction, pages=315–324, year=2006, publisher=Springer

# 8 Appendix