Abstract

text here

1 Introduction

This project focuses on classifying digits from street view images. We use the Street View House numbers dataset from [1]. All images have a fixed 32 32 resolution with character-level ground truth labels. For each example, the labelled character is centered at the image. Since this is a digit recognition task there are ten classes in total.

The data is collected from street-view images, thus there exist vast intra-class variations. To generate competitive performance, we have considered a variety of classification techniques exploiting feature representations that are robust to intra-class variations. We attempt both hand-crafted and learned features as well as raw pixel data as the purpose of this project is to investigate different machine learning techniques and so knowing the pitfalls and prevalences of each technique.

2 Techniques

2.1 Convolutional Neural Networks

2.2 K-Nearest Neighbours

K-Nearest Neighbours (K-NN) has traditionally performed very well in a variety of tasks so we will investigate the performance of K-NN in the SVHN domain. Using the raw pixel data ($32x32x3 = 3072$ dimensions) was too time intensive and so PCA was used to reduce the dimensionality to make the computations more feasible. The PCA dimension space was determined through cross-validation as well as inspecting the eigenvector weights to determine the significance each eigenvector has for preserving the information contained in the dataset (see Section 2.2.1 for the direct analysis on top PCA dimensions). Additionally, the number of neighbours to consider when determining the class for a new example (K hyper-parameter) was also determined using cross validation. **Finally, the tuned K and the PCA dimension hyper-parameters (PCA=3, K=7 chosen from Figure 1) were trained with the "extra_train" dataset to realize the optimal performance for K-NN in the SVHN domain given our data sets. The non-parametric classifier produced an accuracy of %.(this takes too long..still want to do)**

2.2.1 Determining PCA Dimension

The top 20 eigenvector weights were: [0.58293515, 0.05988745, 0.05289433, 0.04385749, 0.02023858, 0.01697088, 0.01630875, 0.01405416, 0.01276369, 0.01054048, 0.00940606, 0.00807162, 0.00638873, 0.00593508, 0.00566315, 0.00521857, 0.00484299, 0.00470581, 0.00434406, 0.00396934]. From inspecting this list, the first eigenvector is extremely influential in separating the classes and after the 4th eigenvector the difference in significance is negligible so we expected either negligible or no performance increase. Therefore, we used PCA projected dimension spaces of $\{1, 2, 3, 4, 30\}$ to see the impact across the top eigenvectors as well as seeing if using additional, seemingly negligible, eigenvectors can have any impact.

2.2.2 Determining K

Given the problem is a ten-ary classification problem it seems natural that K should be fairly large, larger than a configuration for a binary classifier which ranges K from 1 to ≈ 15 (in most domains). Therefore cross validation is executed for all $K \in \{1, 7, 13, 19, 25, 31, 37, 43, 49, 55, 61\}$. The choice of K is not a linear function because, although there are more classes, it should be that each class has a few classes that it may be similar to and others that are almost complete opposites. For example, 1 will be similar to 7 and dissimilar others, 3 can be similar to 8 and 9, while 0 can be similar to 3, 8, and 9 by using common shapes and similar features like roundness and vertical and horizontal displacement. Figure 1 illustrates that the optimal K was always less than $K = 31$.

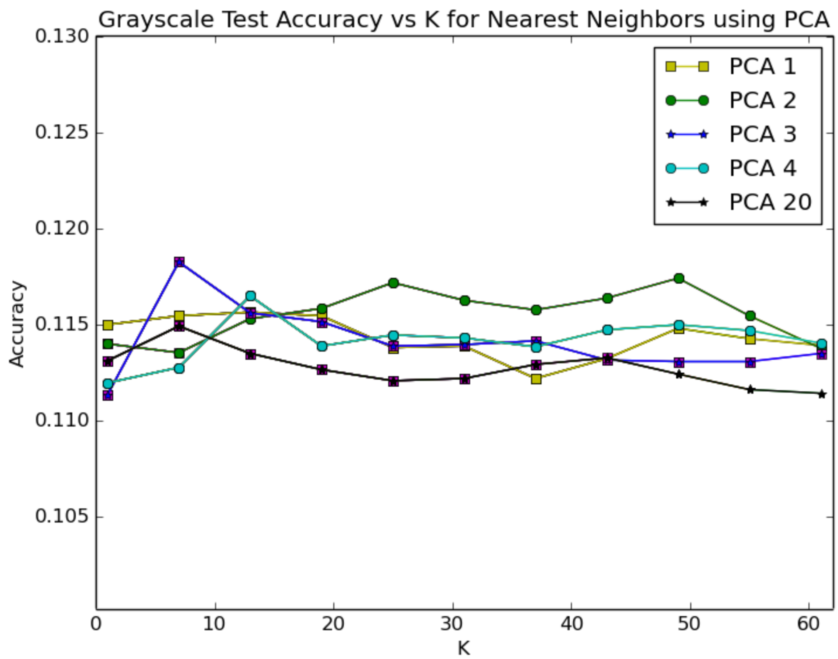

Figure 1: Performance of K-NN grayscale images projected into a Different Dimensional Spaces using PCA

2.2.3 Grayscale PCA Pixel Values

Figure 1 shows the accuracy of using K-NN to classify SVHN using varying PCA dimension spaces and values for K. The accuracies range from $\approx$ 11.1% to $\approx$ 11.8%; however, all are very low suggesting that looking at raw grayscale pixel values is not indicative of interpreting SVHNs (especially considering the extra redundancy in choosing extra PCA eigenvectors). These results are only slightly better than random (10%). In conclusion, K-NN is clearly an extremely weak classifier for SVHN and is likely because of the nature of a colour/3-channel/high-dimensional input example coupled with a ten-class classification problem. Even using grayscale inputs reduced the dimensions significantly, there is likely a lot of information that is lost (otherwise the accuracies would be much higher). The use of PCA was successful in further reducing the grayscale image dimension as seen in Figure 1 which clearly shows that using the top 3 dimensions is sufficient and optimal (given our dataset).

2.2.4 Concluding Remarks

3 Conclusion

Acknowledgments

References

References

[1] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.