Optical Character Recognition

Approach using k-nearest neighbors and Support vector machines

Alla Marchenko, Olav Markussen and Geir Kulia

April 23, 2017

1 Feature Engineering

In this section, the preprocessing of the images are shown. The preprocessing is preformed to enable easier classification for the classifier. As two different classifier approaches are examined in this report, it will also explain two different setups with feature engineering.

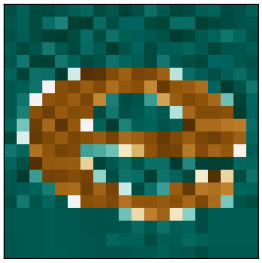
1.1 Feature engineering for k-Nearest Neighbor Classifier

Several different preprocessing tools were explored to make the character easier to recognize for the classifier. The first preprocessing tool applied was total-variation denoising on n-dimensional images [1]. This technique reduces the total variation of an image. This is useful to avoid classifying noise, as the picture will now contain fewer components with high spacial frequency. An example of this denoising technique is shown in the image Figure 1b.

To increase contrast, we used Otsu's method, which is a way to perform image thresholding based on clustering [2] automatically. After thresholding, the image was morphologically closed. This is a standard technique in image processing to remove small wholes in binary images. The result is shown in Figure 1c. All though the result was visually pleasing; it did not decrease the error rate. Therefore, this method was discarded from further analysis.

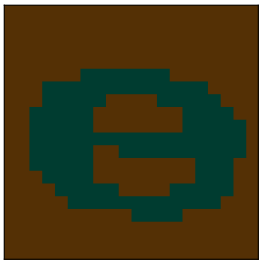
1.1.1 Principal component analysis

It is a substaintian challange and computationally heavy to preform classification on high dimentional data. Therefore, a dimention reduction tool is preferable before performing a k-Nearest Neighbor Classifier. In this project, the Principal Component Analysis procedure was used for reducing the number of correlated dimentions down to a set of linearly uncorrelated principle component. The optimal condition for the k-Nearest Neighbor Classifier is when PCA is used to reduce the dimention down to 57.

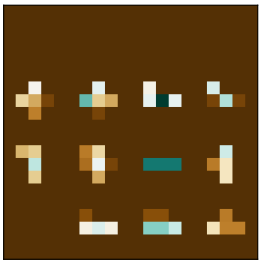


(a) Raw image.

(b) Denoised image.



(c) Binary image.



(d) Histogram of Oriented Gradients.

Figure 1: Example of random selected image before and after preprocessing. The colors has been altered using a color map to visually enhance variance.

References

[1] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging*, 2004.

[2] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 1979.