A Comparison of Classifications in Handwritten Digit Recognition – 25 % Report

Siyang Xue and Tiancheng Liu

1 Problem Description

Digit recognition has been a wide-spread and high-impact pattern recognition task, due to its simplicity: the background and the foreground (digits) are often well-separated, and the classes (categories) are limited to 10 (10 digits). Many researchers have benchmarked different methods on those datasets. Y. LeCun et al.[4] extracted their MNIST dataset from the NIST dataset, and made it normalized and available on their webpage[1]; they also provided early works on different classifications configurations in 1995, where linear classifiers, k-NN classifiers, and multiple variation of neural networks are investigated; a more complicated and complete comparisons can be found on the same webpage. A more recent work by CL Liu et al.[5] benchmarked methods of wider range on multiple datasets, on both feature extraction methods and classification methods.

What we are proposed is to further and deeper investigate the aspects on the digit recognition task. What we care about is how the different options in the classification pipeline affect the performance: *a*) How different normalization techniques influence the data quality, and how to provide a measurement (or at least, an intuition) on that; *b*) how SIFT[6] and other feature extraction methods perform on feature extraction/generation; *c*) what is the suitable dimension reduction method for the task; *d*) how the classifiers and the parameter setting affect the performance of the task.

2 Data Description

The data we used are from 2 different digit dataset, which are both well-known and widely-used for digit recognition:

a) MNIST data[4], which is the most-popular dataset for the task, all data are normalized beforehand;

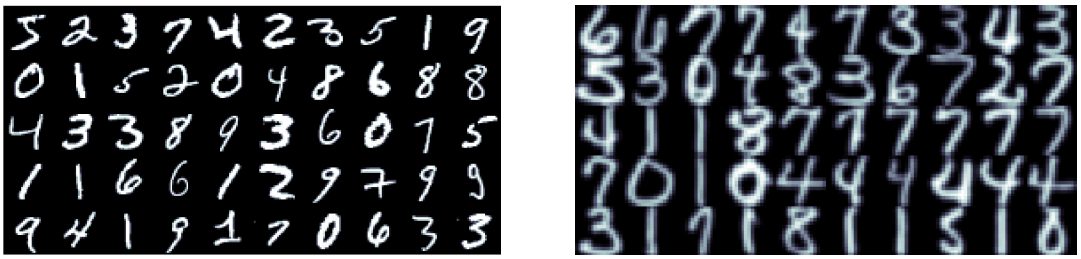
b) USPS handwritten digits data[3], where the data was not normalized, and the testing data is harder than the training data.

Some important information of the datasets are listed as follows:

a) MNIST: Total training set of 60,000 images; test set of 10,000 images; all the digits are normalized to size 20×20 and put in the middle of a 28×28 image.

b) USPS: 7291 for training and 2007 for testing; each image has 300 pixels (10×30).

[1] `http://yann.lecun.com/exdb/mnist/`



(a) (b)

Figure 1: Datasets: (a) MNIST (b) USPS

In this report, we only work on a subset of MNIST dataset, where 6000 digits as training sample, and another 1000 digits as testing data.

All the training and testing samples are drawn respectively from the original training set and testing set. For training, we drawn 600 digits from each subject (0-9) randomly; 100 digits are drawn from each subject from the testing set to gather the subset of testing samples.

3 Methods

3.1 Preprocessing

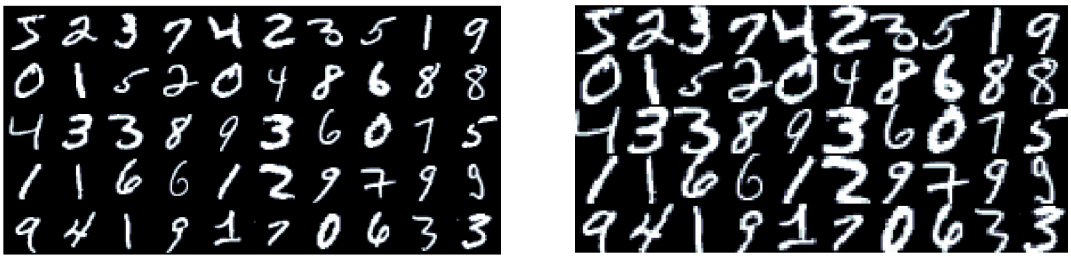
The preprocessing step includes 3 sub-steps: normalization, feature extraction/generation, and dimension reduction.

3.1.1 Normalization

The normalization process removes the size variation and illumination variation of the digits. For the MNIST dataset, as suggested by [4], we removed the padding area to shrink each image from 28×28 to 20×20, which made the matching distance metric more precise and reduced the computational cost of further analysis. All the intensity data are normalized by the maximum intensity in the image, which removes the illumination variance between different images of digits. The result of the normalization can be seen in Figure 3.

3.1.2 Feature Extraction

Feature extraction/generation methods based on gradient and local invariant feature detectors are investigated. To be specific, in this report, we investigated two common feature extraction strategy in the Computer Vision society: a) HOG features and b) Scale-invariant feature transform (SIFT) features.



(a) (b)

Figure 2: Normalization of the dataset: (a) MNIST(original) (b) MNIST(normalized)

HOG Features

Histograms of Oriented Gradients(HOG)[1][2] is a feature selection method based on detection of local gradient distributions. Not only detection of edge positions, but also intensity normalization is included in this method, making it invariant to the shadows or illumination difference of the sample image. A lot of HOG implements are proposed, with different parameter and slightly differently designed histogram calculating schemes. In this project, we will focus mainly on a Felzenszwalb et al. version[2] of HOG, for its relatively fewer features in selecting, as we only care about the distinguish from 0 to 9 in handwritten digits recognition, and too many features could be disturbing.

First, in order to identify the local edges orientation and intensity, the image is divided into cells of 4. In each cell, a histogram of the gradient orientation is created by dividing the orientations θ of the gradient evenly into 9 bins, and each vote has the weight of r(i.e. the magnitude of the coordinate pixel). Therefore, for a $w \times h$ image, a $w/4 \times h/4 \times 9$ cell feature map C is created.

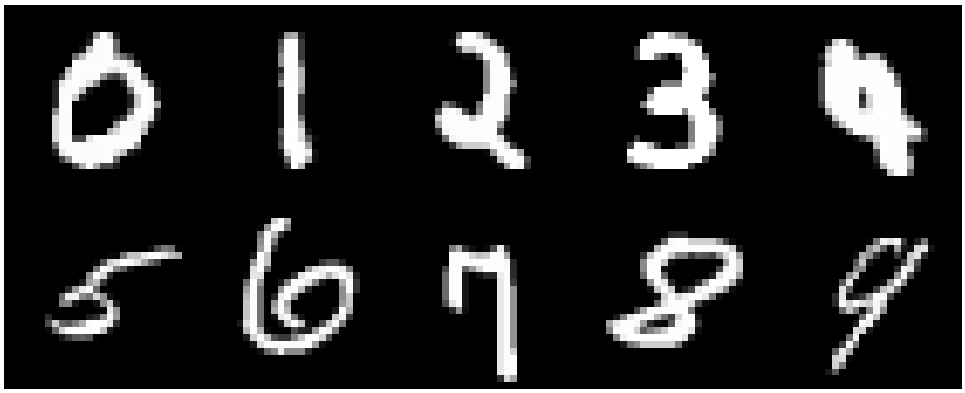
Then, for each items of the feature map $C(i, j, :)$, it is normalized with its 8-neighbor items by creating a 4×9 matrix.

$$HOG(i,j,:,:) = \begin{bmatrix} C(i,j,:)./N_{-1,-1}(i,j,:) \\ C(i,j,:)./N_{-1,+1}(i,j,:) \\ C(i,j,:)./N_{+1,+1}(i,j,:) \\ C(i,j,:)./N_{+1,-1}(i,j,:) \end{bmatrix}$$

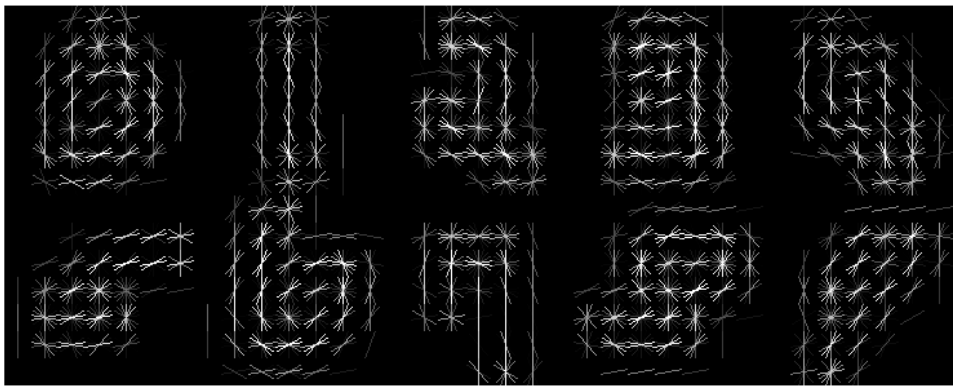
$$N_{\delta,\gamma} = \left(||C(i,j,:)||^2 + ||C(i+\delta,j,:)||^2 + ||C(i,j+\gamma,:)||^2 + ||C(i+\delta,j+\gamma,:)||^2\right)^{\frac{1}{2}} \tag{1}$$

[2] Each matrix is then reshaped into a 36 dimensional vector, resulting a overall $w/4 \times h/4 \times 36$ feature map.

Finally, the now normalized feature map is compressed to 31 features per cell, using PCA projecting the features to 31 most variant dimensions in the $w/4 \times h/4$ features of the image. This step is different from the later dimension-reduction step of all the images in the training set in that it calculates the principal components within the same image and is aimed at eliminating the specific minor noises of a particular image.



(a)



(b)

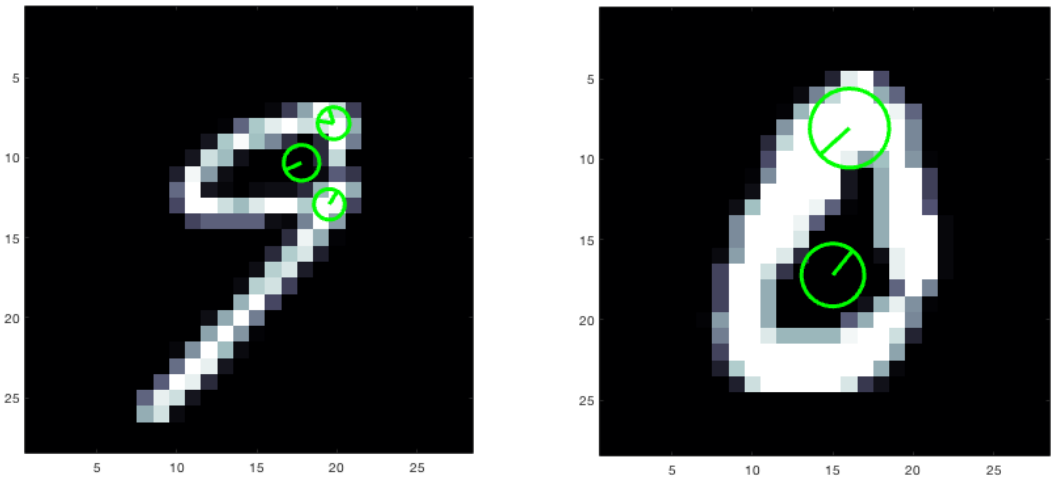
Figure 3: (a)handwritten digits 0-9 from MNIST (b)HOG features of handwritten digits 0-9

The original images, and the HOG features of digits from 0 to 9 is shown in Figure 1 and Figure 2 respectively. As shown in the figures, the HOG features well describe the outlines of the digits. A digit written with a thinner line(such as the '9' in Figure 1) does not make much difference in the HOG feature description with digits written with thicker strokes, making the features resistant to the irrelevant factors.

SIFT Features

Scale-invariant feature transform is a popular feature extraction technique introduced by David Lowe in his 2004 paper [6]. The process consists of typically three steps: a) Detection of scale-space extrema, where image pyramid is generated using a set of different-scaled Gaussian kernels, and the extrema is detected in a spatial-hierarchical neighborhood; b) Accurate keypoint localization, where the extrema of low contrast and extrema near the image boundary are discarded; c) Orientation assignment, where each keypoint is assigned several orientations according to its spatial neighbors.

The SIFT features are often applied in object detection, face recognition, and other



(a) (b)

Figure 4: Handwritten digits with their SIFT features: (a) A handwritten 9 (b) A handwritten 0

related tasks [7, 1], due to its robustness against image scaling, image rotation, and certain kinds of noise. In these applications, the SIFT features (keypoints) are extracted, and oftentimes, it is followed by extracting the HOG features of the spatial neighborhood of the keypoints, to generate a feature descriptor [6].

However, this feature extraction strategy is ineffective for the MNIST dataset. In the experiment, we found all the images have only a few feature points to be extracted (≤ 4), which is insufficient to describe the image (see Figure 4); some image even failed on SIFT, that means it have no SIFT feature to extract. Furthermore, it may not be a desirable choice for digits dataset, where we have 6 and 9 with the same topology, and the rotation invariant property of SIFT would made it impossible to distinguish the above 2 digits. Therefore, in this report, we do not use SIFT as a feature extraction technique, and only HOG is used to generate feature descriptors.

However, for further study, if we are to investigate faces, SIFT would be a good choice for extracting the keypoints in a more complex context, and the the invariant properties of the SIFT would help the recognition.

3.1.3 Dimension Reductions

The dimension reduction methods used in this report can be categorized into 2 groups: linear and nonlinear. For linear method, we apply the conventional PCA; both kernel method (Kernel-PCA) and manifold learning method (LLE) are investigated as nonlinear methods. Experiments are conducted to compare the effectiveness and efficiency of the methods.

3.2 Clustering

After the preprocessing, we plan to investigate clustering methods on the new feature space. The clustering methods such as k-means, EM and DBSCAN will be investigated to show the intrinsic nature of the dataset, and will be used as references to justify and validate the further classification process. This part is to be discussed in the future report.

3.3 Classification

Different classification methods are to be investigated in the final stage. In this report, we used kNN classifier for simplicity to appreciate the techniques introduced in the former sections.

4 Results

In this section, we discuss the differences of the performances of the classifier with input types: features without selection, HOG features selected with raw data, and further HOG features selected with normalized data over the several dimensional reduction methods. Running time and classification accuracy are both included in the measurement of the performance.

Table 1: Accuracy and Running Time Without Feature Selection or Normalization

	None	**PCA**	**Poly**	**Gaussian**	**LLE**
Accuracy(%)	87.80	90.20	90.10	91.90	91.90
Running Time(s)	–	5.72	76.68	80.20	38.95

Table 2: Accuracy and Running Time With Normalization and Without Feature Selection

	None	**PCA**	**Poly**	**Gaussian**	**LLE**
Accuracy(%)	93.00	93.30	92.60	94.00	95.10
Running Time(s)	–	0.61	283.50	295.91	18.71

Table 3: Accuracy and Running Time Without Normalization and With Feature Selection

	None	**PCA**	**Poly**	**Gaussian**	**LLE**
Accuracy(%)	93.90	95.90	95.20	96.20	96.50
Running Time(s)	14.70	17.01	99.61	107.39	101.39

Table 4: Accuracy and Running Time With Normalization and With Feature Selection

	None	**PCA**	**Poly**	**Gaussian**	**LLE**
Accuracy(%)	95.40	95.70	95.10	95.90	96.50
Running Time(s)	7.24	7.98	92.52	101.76	45.01

4.1 Comparison on Normalization

Comparing Table 3 with Table 4, the accuracy of features with normalization previous to dimension reduction is not higher than that of features without normalization. That is partly because of the information loss in the gradient near the frame of the digits. On the contrary, normalization performed with no dimension reduction helps to improve the accuracy of the classification by eliminating irrelevant part of the features, as there is no dimension reduction process selecting the major features here. At the same time, normalization will decrease the dimension of features, thus decreasing the running time of dimension reduction method.

4.2 Comparison on Feature Selection

By comparing Table 3 with Table 1, and Table 4 with Table 2, basically a increase of accuracy of 4% to 7% is achieved by selecting HOG features. Better edge detection and shadow invariant property of the HOG features might contribute to this increase. The increase is especially large in the case of none dimension reduction (7.6%) as the classification performance is more dependent to the feature quality with no further dimension reduction step. The running time is longer with feature selection, as the original dimension of the data is 784, and the selected feature dimension here is 775, so there is no much decrease in the dimension reduction process, and the increase in computational complexity comes from the HOG feature detection step. This increase, however, is minor(less then 10 percent) compared to the accuracy improvement.

5 Conclusion

In this report, we investigated the normalization method, feature selection methods, and dimension reduction methods, with a simple kNN classifier. Several experimental comparisons are also made to appreciate the effect of each step. The conclusions are as follows: a) HOG feature are used for feature selection, and is helpful for digit recognition; b) SIFT is not suitable for digit recognition, due to the simple structure of the digit images, and the topology similarity too high between the digits can make it impossible to distinguish the digits from others; c) Dimension reduction methods help the recognition, and the nonlinearity of the dataset is demonstrated.

For next step, we would begin to consider clustering methods and different classification methods.

References

[1] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[2] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[3] Jonathan J Hull. A database for handwritten text recognition research. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(5):550–554, 1994.

[4] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[5] Cheng-Lin Liu, Kazuki Nakashima, Hiroshi Sako, and Hiromichi Fujisawa. Handwritten digit recognition: benchmarking of state-of-the-art techniques. *Pattern Recognition*, 36(10):2271–2285, 2003.

[6] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[7] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13, 2006.

Tiancheng's contribution

- Set up the Github repository for collaboration
- Normalized the MNIST dataset (max-normalization & depadding)
- Investigated the SIFT, and analysis why it is not a desirable choice for digit recognition.
- Conducted the experimental comparison of normalized v.s. original, based on Siyang's HOG feature extraction
- Fixed the LLE code

Siyang's contribution

- Investigated the HOG, and discussed how it improved the performance of classification by detecting edges more accurately
- Running the experimental comparison between the feature selected and none feature selected classification.
- Fixed the kernel-PCA code