Color Image Classification Using Locally Linear Manifolds and Learning

Kazuki KONDO
Tokyo University
of Agriculture and Technology
email : k.kondo.0823@gmail.com

Seiji HOTTA
Tokyo University
of Agriculture and Technology
email : seiji.hotta@gmail.com

Abstract

In this paper, we propose block matching and learning using linear manifolds (affine subspaces) for color image classification. In our method, training images are partitioned into small size blocks. Given a test image, it is also partitioned into small size blocks, and a linear manifold corresponding to each test block is formed by its neighbor training blocks. Our method classifies a test image into the class that has the shortest total sum of the projection distances between test blocks and their corresponding linear manifolds. We also propose a learning algorithm for reducing the number of blocks without accuracy deterioration. Experimental results show that our classification performs as well or better than other classifiers such as support vector machine with bag-of-keypoints.

1 Introduction

Object recognition is one of the most challenging problems in computer vision. For this task, *bag-of-keypoints* was proposed by Csurka *et al.* [1] and developed by many researchers. This approach is based on vector quantization of affine invariant descriptors of image patches, and images are represented as cluster co-occurrence histograms. By using these histograms, images are classified with *Support Vector Machine* (SVM). Since this method can achieve high accuracy for object recognition, many researchers have developed it [2, 3, 4, 5]. However, the main drawback of these methods is that landscape images such as sea and mountains are not classified correctly because it is difficult to extract effective local features for classification from such images. Moreover, the learning process of SVM will not converge when the numbers of training samples and classes are large. This drawback will be a critical problem in near future because the numbers of training samples and classes in object recognition are growing increasingly.

For overcoming these difficulties, color image classification using block partition has been proposed in 2007 [6]. In the method, all training images are partitioned into small block images (cf. Fig. 1 (a)). Given a test image, it is also partitioned into small blocks. After block partition, k neighbor training blocks corresponding to each test block are selected in each class (cf. Fig. 1 (b)). In a classification phase, mean blocks are calculated with selected neighbor blocks in each class, and the total sum of distances between test blocks and their corresponding mean ones is measured in individual classes. This total sum of distances can be regarded as the distance between a test image and images reconstructed by mean blocks (cf. Fig. 1 (c)). Finally, a test image is classified into the class that has the shortest total sum of distances between test and mean blocks. This method does not use local features and SVM, so we do not suffer from the difficulties described above. However, classification cost and memory requirement of this approach tend to be large because the method is a kind of memory-based classifiers. The simplest way of reducing classification cost and memory requirement is to cut down the number of training blocks, but such reduction yields accuracy deterioration because the representation capacity of mean blocks is very limited.

In this paper, we propose block matching using locally linear manifolds (affine subspaces) instead of mean blocks for color image classification. In a preprocessing phase, k-neighbor training blocks of a test one is selected in each class, and $(k-1)$-dimensional locally linear manifolds are formed by selected neighbor training blocks. In a classification phase, a test image is classified into the class that has the shortest total sum of the projection distance between each test block and its corresponding locally linear manifold. By using linear manifolds, it is expected that we can expand the representation capacity of an available small number of training blocks, so we can reduce the number of training blocks easily. However, such locally linear manifolds formed by k neighbor training blocks do not guarantee high accuracy. Hence, we apply a learning rule based on generalized learning vector quantization (GLVQ) [7] to locally linear manifolds for improving accuracy. The performance of our color image classification is verified with experiments on the WANG color image dataset [8].

2 Related work

The most related work to our color image classification is [6], so in this section, let us start with a brief review of the method described in [6]. Let $\mathbf{X}_i^{(j)}(i = 1, ..., N_j)$ be the ith arbitrary size training image belonging to class $j(j = 1, ..., C)$, where N_j and C are the numbers of images belonging to class j and classes, respectively. In a preprocessing phase, all training images are partitioned into block images of which sizes are $m \times n$ pixels. Assume n_j training blocks are obtained from class j by this partitioning. Let $\boldsymbol{x}_i^{(j)}(i = 1, ..., n_j)$ be the vector of which elements are pixel values of the ith training block. In color image classification, color pixel values are directly used as feature vectors. Consequently, the dimensionality of $\boldsymbol{x}_i^{(j)}$ is $m \times n \times 3$. By using color features, we can easily classify images that are difficult to describe by local features such as landscape images.

When an arbitrary size test image $\mathbf{Q}$ is given, partition it into block images of which sizes are $m \times n$ pixels. Assume n_q test blocks are obtained from $\mathbf{Q}$ by this partitioning. Let $\boldsymbol{q}_l(l = 1, ..., n_q)$ be the vector of which elements are pixel values of the lth test block of $\mathbf{Q}$. Let $\mathcal{N}_l^{(j)}$ be the set of k closest training blocks of $\boldsymbol{q}_l$ which are selected from class j using Euclidean distance $d(\boldsymbol{q}_l, \boldsymbol{x}_i^{(j)}) = \|\boldsymbol{q}_l - \boldsymbol{x}_i^{(j)}\|^2$. After that, the mean block of the selected neighbor training blocks is calculated by

$$\boldsymbol{m}_l^{(j)} = \frac{1}{k} \sum_{i \in \mathcal{N}_l^{(j)}} \boldsymbol{x}_i^{(j)}. \quad (1)$$

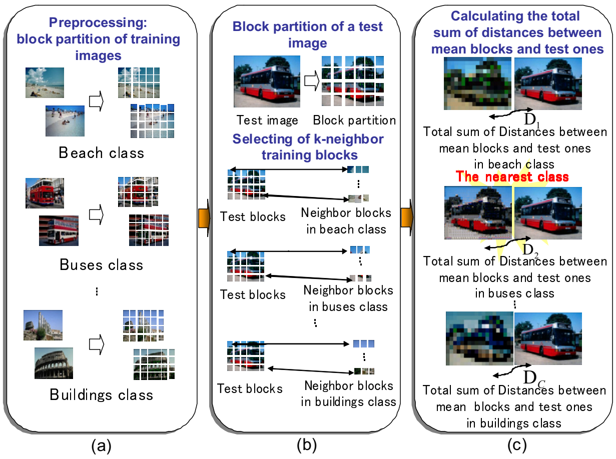

Figure 1: Color image classification using block partition.

We calculate n_q mean blocks $\boldsymbol{m}_1^{(j)}, ..., \boldsymbol{m}_{n_q}^{(j)}$ corresponding to each test block $\boldsymbol{q}_l (l = 1, ..., n_q)$ in individual classes. For a classification phase, the total sum of distances between $\boldsymbol{q}_l$ and $\boldsymbol{m}_l^{(j)}$ is calculated by

$$\bar{D}_j = \sum_{l=1}^{n_q} d(\boldsymbol{q}_l, \boldsymbol{m}_l^{(j)}). \tag{2}$$

The above $\bar{D}_j$ is equivalent to the distance between a test image and an image reconstructed by mean blocks in class j. Hence, a test image $\mathbf{Q}$ is classified into the class that has the minimum $\bar{D}_j$, i.e., the class of $\mathbf{Q}$ (denoted as ω) is determined by the following classification rule:

$$\omega = \arg \min_{j=1,...,C} \bar{D}_j. \tag{3}$$

Classification rules using a mean vectors of k nearest training samples are called *local mean-based classifier* (LMC) [9, 10]. Hence, we call the above image classification method *LMC-based method* for short. The LMC-based method is a kind of memory-based classifiers, so it requires a large amount of blocks and high classification cost.

3 Color image classification using locally linear manifolds

The simplest way of reducing classification cost and memory requirement of the LMC-based method is to cut down the number of training blocks. However, such reduction yields accuracy deterioration due to a small representation capacity of mean blocks. By using linear manifolds, it is expected that we can expand the representation capacity of available small number of training blocks. As an example, a two-dimensional linear manifold spanned by three training blocks is shown in Fig. 2. Each of the corners of the triangle represents pure training blocks, whereas the gray area in between represents linear combinations of them. These intermediate training samples can be used as artificial training blocks for classification. Due to this property, manifold-based classifiers tend to outperform mean-based ones in high-dimensional pattern classification. In addition, we can reduce the classification cost and memory requirement of manifold-based classifiers easily compared with mean-based ones.

For measuring the distance between the lth test block $\boldsymbol{q}_l$ and the locally linear manifolds spanned by $\mathcal{N}_l^{(j)}$, solve the following optimization problem [10]:

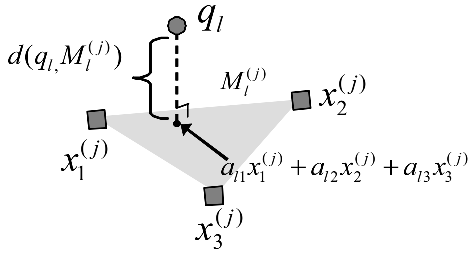

Figure 2: Example of a locally linear manifold.

$$\begin{aligned} \min_{\boldsymbol{a}_l} \quad & \|\boldsymbol{q}_l - \sum_{i \in \mathcal{N}_l^{(j)}} a_{li} \boldsymbol{x}_i^{(j)}\|^2 + \lambda \|\boldsymbol{a}_l\|^2 \\ \text{s.t.} \quad & \boldsymbol{a}_l^\top \mathbf{1}_k = 1, \end{aligned} \tag{4}$$

where $\boldsymbol{a}_l = (a_{l1} \cdots a_{lk})^\top \in \mathbb{R}^k$ is a weight vector for the linear combination of k neighbor training blocks from class j, and $\mathbf{1}_k = (1 \cdots 1)^\top \in \mathbb{R}^k$ is a vector of which all elements are 1. In addition, λ is a regularization parameter. According to our experiments, this parameter is not sensitive to recognition accuracy, so λ was fixed to $\lambda = 1.0 \times 10^{-2}$ in our experiments. The same cost function can be found in the first step of locally linear embedding [11]. The optimal weights subject to sum-to-one are found by solving a least-squares problem. The solution of the above constrained minimization problem can be given in closed form by using Lagrange multipliers. In brief, the optimal weight $\boldsymbol{a}_l$ is given as follows:

$$\boldsymbol{a}_l = \frac{(\mathbf{C}_l^\top \mathbf{C}_l + \lambda \mathbf{I})^{-1} \mathbf{1}_k}{\mathbf{1}_k{}^\top (\mathbf{C}_l^\top \mathbf{C}_l + \lambda \mathbf{I})^{-1} \mathbf{1}_k}, \tag{5}$$

where $\mathbf{C}_l$ and $\mathbf{I}$ represent a $d \times k$ matrix $\mathbf{C}_l = (\boldsymbol{q}_l - \boldsymbol{x}_1^{(j)} | \cdots | \boldsymbol{q}_l - \boldsymbol{x}_k^{(j)})$ and a $k \times k$ identity matrix, respectively. Of course, $\boldsymbol{x}_i^{(j)}$ $(i = 1, ..., k)$ in $\mathbf{C}_l$ is an element of $\mathcal{N}_l^{(j)}$. By using the above optimal weight, the distance between $\boldsymbol{q}_l$ and the linear manifold spanned by $\mathcal{N}_l^{(j)}$ (denoted by $\mathcal{M}_l^{(j)}$) can be measured as follows:

$$d(\boldsymbol{q}_l, \mathcal{M}_l^{(j)}) = \|\boldsymbol{q}_l - \sum_{i \in \mathcal{N}_l^{(j)}} a_{li} \boldsymbol{x}_i^{(j)}\|^2 + \lambda \|\boldsymbol{a}_l\|^2. \tag{6}$$

We calculate n_q linear manifolds $\mathcal{M}_1^{(j)}, ..., \mathcal{M}_{n_q}^{(j)}$ corresponding to each test block $\boldsymbol{q}_l (l = 1, ..., n_q)$ in individual classes. After that, the total sum of distances between $\boldsymbol{q}_l$ and $\mathcal{M}_l^{(j)}$ is calculated by

$$D_j = \sum_{l=1}^{n_q} d(\boldsymbol{q}_l, \mathcal{M}_l^{(j)}). \tag{7}$$

The above D_j can be regarded as the minimum distance between a test image and an image reconstructed by locally linear manifolds in class j. In a classification phase, a test image $\mathbf{Q}$ is classified into the class that has the minimum D_j, i.e., the class of $\mathbf{Q}$ is determined as follows:

$$\omega = \arg \min_{j=1,...,C} D_j. \tag{8}$$

The above rule is our proposed color image classification rule. Incidentally, a classification rule using a linear manifold spanned by k nearest training samples of a test sample is called *local subspace classifier* [10, 12].

4 Computational cost reduction

By using locally linear manifolds, the representation capacity of an available small number of training blocks is expanded, so we can decrease the number of training blocks. However, a small number of training blocks does not guarantee high accuracy. For overcoming this difficulty, we apply a learning rule based on generalized learning vector quantizastion (GLVQ) [7] to locally linear manifolds for improving accuracy. In GLVQ, prototypes called reference (codebook) vectors are updated by a steepest descent method that minimizes a cost function defined by the distance between a test vector and its nearest prototype. However, we cannot apply GLVQ to our method directly because our classification rule is defined with the total sum of distances between each test block and its corresponding linear manifold. Hence, we change the cost function of GLVQ for our classification, i.e., we define the cost function by using the total sum of projection distances. Consequently, the update rule of our learning is different from the original GLVQ algorithm, i.e., in our learning, all neighbor blocks are updated into optimal positions for improving accuracy.

4.1 Block reduction with learning

In the learning rule shown in here, d-dimensional vectors $\boldsymbol{p}$ called *reference block* are optimized using training blocks. Initial reference blocks are selected from training blocks of each class randomly. These reference blocks then are optimized using training images.

Let $\mathbf{X}$ be a training image, and $\boldsymbol{x}_l$ $(l = 1, ..., n_x)$ represents the lth block of $\mathbf{X}$. Here, the locally linear manifold for the lth training block $\boldsymbol{x}_l$ belonging to the same class of $\mathbf{X}$ is denoted as $\mathcal{M}_l^{(1)}$ $(l = 1, ..., n_x)$. In contrast, the linear manifold for $\boldsymbol{x}_l$ belonging to the different class from $\mathbf{X}$ is denoted as $\mathcal{M}_l^{(2)}$. Let $\mathbf{P}_l^{(1)} = (\boldsymbol{p}_{l1}^{(1)}, \cdots, \boldsymbol{p}_{lk}^{(1)})$ be the k-nearest reference blocks that spans $\mathcal{M}_l^{(1)}$. In contrast, let $\mathbf{P}_l^{(2)}$ be the k-nearest reference blocks that spans $\mathcal{M}_l^{(2)}$. Let us here define the relative distance difference $\mu(\mathbf{X})$ for our classification as follows:

$$\mu(\mathbf{X}) = \frac{D_1 - D_2}{D_1 + D_2}, \tag{9}$$

where $D_1 = \sum_{l=1}^{n_x} d(\boldsymbol{x}_l, \mathcal{M}_l^{(1)})$ and $D_2 = \sum_{l=1}^{n_x} d(\boldsymbol{x}_l, \mathcal{M}_l^{(2)})$ (cf. Eq.(6)). The above $\mu(\mathbf{X})$ satisfies $-1 < \mu(\mathbf{X}) < 1$. If $\mu(\mathbf{X})$ is negative, $\mathbf{X}$ is classified correctly; otherwise, $\mathbf{X}$ is misclassified. For improving accuracy, we should minimize the following cost function:

$$S = \sum_{i=1}^{N} f(\mu(\mathbf{X}_i)), \tag{10}$$

where N is the number of training images, and $f(\mu)$ is a nonlinear monotonically increasing function. To minimize S, we adopted a steepest descent method with a small positive constant $\alpha (0 < \alpha < 1)$ to Eq. (10):

$$\mathbf{P}_l^{(j)} \leftarrow \mathbf{P}_l^{(j)} - \alpha \frac{\partial S}{\partial \mathbf{P}_l^{(j)}}, \ (j = 1, 2), \tag{11}$$

where $\partial S / \partial \mathbf{P}_l^{(j)}$ is derived as

$$\frac{\partial S}{\partial \mathbf{P}_l^{(j)}} = \frac{\partial S}{\partial \mu} \frac{\partial \mu}{\partial D_j} \frac{\partial D_j}{\partial \mathbf{P}_l^{(j)}},$$
$$= (-1)^j \frac{\partial f}{\partial \mu} \frac{4 D_{3-j}}{(D_1 + D_2)^2} (\boldsymbol{x}_l - \mathbf{P}_l^{(j)} \boldsymbol{a}_l^{(j)}) \boldsymbol{a}_l^{(j)\top}, \ (j = 1, 2).$$

Figure 3: Image examples of the WANG dataset.

In the above equation, $\boldsymbol{a}_l^{(j)}$ represents the weight vector for $(\boldsymbol{p}_{l1}^{(j)}, \cdots, \boldsymbol{p}_{lk}^{(j)})$ determined by Eq. (5). Consequently, the update rule for our image classification is given as follows:

$$\mathbf{P}_l^{(j)} \leftarrow \mathbf{P}_l^{(j)} - \delta_l^{(j)} (\boldsymbol{x}_l - \mathbf{P}_l^{(j)} \boldsymbol{a}_l^{(j)}) \boldsymbol{a}_l^{(j)\top}, \ (j = 1, 2),$$

where $\delta_l^{(j)} = (-1)^j \alpha \frac{\partial f}{\partial \mu} \frac{D_{3-j}}{(D_1 + D_2)}$ $(j = 1, 2)$. In our experiments, $f(\mu, t)\{1 - f(\mu, t)\}$ is used for $\partial f / \partial \mu$, where t is learning time and $f(\mu, t)$ is a sigmoid function $1/(1 + e^{-\mu t})$. In a learning phase, k nearest blocks $\mathbf{P}_l^{(j)}$ are updated using all training images until training accuracy converges. If all elements of $\boldsymbol{a}_l^{(j)}$ are fixed to $1/\sqrt{k}$, the above process can be regarded as a learning rule for the LMC-based method (cf. next experiments).

5 Experiments

We tested our method on the WANG color image dataset [8], formed by 10 image classes: African people, beach, buildings, buses, dinosaurs, elephants, flowers, horses, mountains, and food. Each class consists of 100 images of which sizes are 80×120 or 120×80 pixels. Thus, the total number of images is 1000. In this dataset, objects occur over cluttered backgrounds as shown in Fig. 3. In experiments, these images were partitioned into block images of which sizes are 10×10 and 20×20 pixels. All classification methods were implemented with MATLAB on a standard PC that has Pentium 2 GHz CPU and 2 GB RAM.

5.1 Classification performance

First, we investigated the classification performance of the proposed method. Recognition rates were estimated by 5-fold cross-validation with varying the number of neighbors k and block sizes. The same value of k was used over all classes. In this experiment, we compared the proposed method with three conventional methods: (i) the LMC-based method [6], (ii) *bag-of-keypoints* [1], and (iii) support vector machine (SVM). For bag-of-keypoints, SIFT features were extracted from gray-scale images and transformed them into 80 visual words. In SVM, normalized RGB color histograms with a size of 64 bins were extracted from each image as feature vectors. For nonlinear mapping, the Gaussian kernel $K(\boldsymbol{x}, \boldsymbol{y}) = e^{-\alpha ||\boldsymbol{x} - \boldsymbol{y}||^2}$ was used in SVM. In this experiment, the block reduction method was not applied to our image classification.

The recognition rates of each method and their standard deviations are summarized in Table 1. Parameters of each method were determined with validation sets. As shown in this table, our method with 10×10 block outperformed the other methods, and the result of bag-of-keypoints was very poor because effective local features were not extracted from the WANG images. Fig. 4 shows recognition rates with respect to the number of neighbor blocks k (block size is 10×10 pixels). As shown in this figure, the recognition rate of our classifier was higher than that of the LMC-based

Table 1: Recognition rates of each method.

method	accuracy [%]
Our method (10×10)	$\mathbf{88.3 \pm 5.8}$
Our method (20×20)	$\mathbf{82.9 \pm 4.5}$
LMC-based method (10×10)	86.9 ± 5.4
LMC-based method (20×20)	82.3 ± 6.9
SVM	78.5 ± 4.2
bag-of-keypoints [1]	41.8 ± 3.4

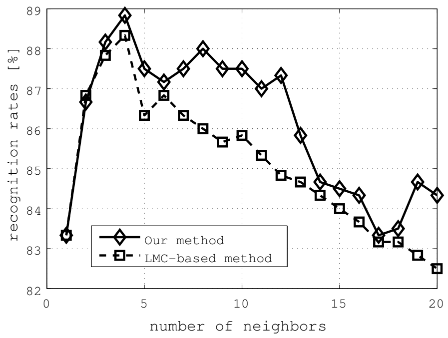

Figure 4: Recognition rates with respect to the number of neighbors k.

method in almost every k. These results show that making use of linear manifolds is effective on color image classification.

5.2 Effectiveness of learning

Finally, we evaluated the effectiveness of learning for our image classification. In this experiment, we selected 20 images from each class randomly for test images. Consequently, the total number of test images was 200. The initial reference blocks were selected from the leftover 800 training images. This random splitting was performed independently in 5-fold cross-validation. Recognition rates were evaluated with varying the number of reference blocks as 10, 25, 50, and 100 per class. For comparison, we also evaluated the LMC-based method with learning. For learning, $\alpha = 10^{-3}$ and $k = 4$ were used as a small positive constant and the number of neighbors, respectively.

Figure 5 shows the recognition rates of our method and the LMC-based method with respect to the number of reference blocks per class (block size is 20×20 pixels). As shown in this figure, even if the number of blocks per class was 10, our method with learning achieved over 75% accuracy, which was the almost same as accuracy of SVM. This result showed that making use of a linear manifold and its learning was effective for improving accuracy of block-based image classification. In addition, our method achieved about 80% accuracy with only 25 reference blocks per class, while the LMC-based method required 50 reference blocks for achieving the almost same accuracy. Table 2 shows classification time of each method. As shown in this table, our classification rule was 1.6 times faster than the LMC-based method for achieving about 80% accuracy. Hence, it can be concluded that we can reduce classification cost and memory requirement by using our method effectively more than the LMC-based method.

6 Conclusion

This paper proposed block matching using linear manifolds (affine subspaces) for color image classification. In our method, a test image is classified into the class that has the shortest total sum of the projection distance between each test block and its corresponding linear manifold. We

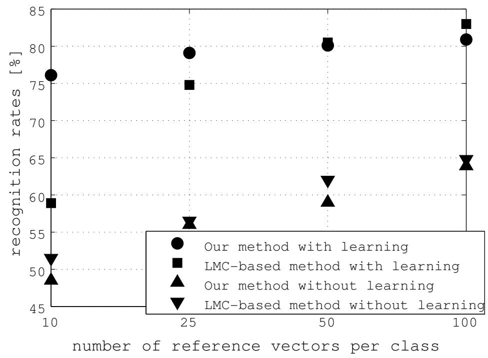

Figure 5: Recognition rates with respect to the number of reference blocks.

Table 2: Classification time (sec) per test image (block size is 20×20 pixels).

♯ blocks	our method	LMC-based method
2160 (all blocks)	23.0	22.9
100	0.91	0.85
50	0.35	0.32
25	0.20	0.19
10	0.12	0.11

also proposed a learning algorithm for reducing the number of blocks without accuracy deterioration. Experimental results showed that our method performed as well or better than other classifiers. Future work will be dedicated for applying object detection to our classification.

References

[1] G. Csurka *et al.*, “Visual categorization with bags of keypoints,” Proc. of ECCV, pp. 1–22, 2004.

[2] A. Bosch *et al.*, “Image classification using random forests and ferns,” Proc. of ICCV, pp. 14–20, 2007.

[3] J. Willamowski *et al.*, “Categorizing Nine Visual Classes using Local Appearance Descriptors,” Proc. of ICPR, pp. 242–249, 2004.

[4] S. Lazebnik *et al.*, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” Proc. of CVPR, pp. 2169–2178, 2006.

[5] K. Grauman *et al.*, “Discriminative classification with sets of image features,” Proc. of ICCV, pp. 1458–1465, 2005.

[6] K. Kondo *et al.*, “Color image classification using block image replacement and local averaging classifier,” Proc. of KJPR, pp. 173–178, 2007.

[7] A. Sato *et al.*, “Generalized learning vector quantization,” Proc. of NIPS, pp. 423–429, 1995.

[8] J.Z. Wang *et al.*, “SIMPLIcity: Semantics-sensitive integrated matching for picture libraries approach” Trans. of PAMI, vol. 23, no. 9, pp. 947–963, 2001.

[9] Y. Mitani *et al.*, “A local mean-based nonparametric classifier,” Patt. Recog. Lett., vol. 27, no. 10, pp. 1151–1159, 2006.

[10] P. Vincent and Y. Bengio, “K-local hyperplane and convex distance nearest neighbor algorithms,” NIPS, 2002.

[11] S.T. Roweis *et al.*, “Nonlinear dimensionality reduction by locally linear embedding,” Science, vol. 290–5500, pp. 2323–2326, 2000.

[12] J. Laaksonen, “Subspace classifiers in recognition of hand written digits,” PhD thesis, Helsinki Univ. of Tech., 1997.