A Spectral Representation for Appearance-Based Classification and Recognition

Xiuwen Liu Anuj Srivastava
Department of Computer Science Department of Statistics
Florida State University, Tallahassee, FL 32306, USA

Abstract

We present a spectral representation for appearance-based image classification and object recognition. Based on a generative process, the representation is derived by partitioning the frequency domain into small disjoint regions. This gives rise to a set of filters and a representation consisting of marginal distributions of those filter responses. We use a neural network to learn a classifier through training examples. We propose a filter selection algorithm by maximizing the performance over training data. A distinct advantage of our representation is that it can be effectively used for different classification and recognition tasks, which is demonstrated by experiments and comparisons in texture classification, face recognition, and appearance-based 3D object recognition.

1. Introduction

With the recent development of sophisticated learning algorithms, it has been realized that the performance of a classification and recognition system critically depends on the underlying representation [1]. Geman and Bienenstock suggested that “the fundamental challenges are about representation rather than learning” [1].

In this paper, we advocate that a priori knowledge should be derived by analyzing the physical generative processes underlying images pioneered by Grenander [2] who proposed pattern theory. The most effective a priori knowledge should then be derived by integrating out factors that do not matter or are too complicated if made explicit. Within this paradigm, we derive a representation for classification and recognition of images based on a simple generative process. The obtained representation, however, has demonstrated to be very effective for a variety of classification and recognition problems that are commonly studied separately in the literature.

2. Spectral Representation

The starting point of our representation is the following scenario. Suppose that we have a large number of different objects which may appear on a uniform background according to a Poisson process and the objects are not known explicitly in any other way. We are interested in a translation invariant statistical feature that can be used to characterize the appearance of images. In other words, the derived feature should be able to classify and recognize the large number of objects based on the observed images.

An obvious choice is the histogram of the given image, which is translation invariant. However, for a large number of objects, their histogram can be very close or even identical, making the histogram not sufficient for recognition and classification. The independence assumption underlying the histogram is not valid as the pixels belonging to one object are dependent. Another obvious choice is to build a joint probability model of all the pixels. However, the dimension of the joint space is too high for computations. Here we seek a compromise between the two extreme choices. Instead of assuming that the pixels are independent, we assume that small disjoint regions in the frequency domain are independent. In other words, we partition the frequency domain into small disjoint regions and model each region by its histogram, or marginal distribution.

How shall we partition the frequency domain? One sensible way is to partition it into rings with a small range of frequencies, as shown in Fig. 1(a). Another way is to partition it into regions with a small range of frequencies and phases, as shown in Fig. 1(b). By using a Gaussian window function, it can be shown that each ring in Fig. 1(a) leads to a difference of Gaussian filter, which can be implemented using a Laplacian of Gaussian filter and each small region in Fig. 1(b) leads to a Gabor filter. While the constructed filters may not be independent, the independence is valid to a certain extent for natural images, as recent numerical studies show that Gabor filters are independent components of natural images [3]. Assuming that their responses are independent, the Kullback-Leibler distance between two joint
Filter and we choose the first to be the one that gives the framework. Initially, we train one neural network for each we seek a numerical procedure within the neural network an analytical solution for filter selection difficult. Instead, see that histograms are nonlinear operators and this makes the performance of the given training data. It is easy to see that the performance is maximized. Our idea is to maximize identical up to a translation.

Figure 1. Two ways of partitioning the frequency domain. (a) Ring structures. (b) Small regions.

distributions is the sum of their corresponding marginal distributions as shown, by the following equation, where \( p_i(x_1, \cdots, x_n) \) is the joint distribution and \( p_i(x_j) \) the \( j \)th marginal distribution:

\[
KL(p_1(x_1, \cdots, x_n), p_2(x_1, \cdots, x_n)) = \int_{x_1} \cdots \int_{x_n} \frac{p_1(x_1, \cdots, x_n)}{p_2(x_1, \cdots, x_n)} \log \frac{p_1(x_1, \cdots, x_n)}{p_2(x_1, \cdots, x_n)} \, dx_1 \cdots dx_n \\
= \sum_{i=1}^{n} \int_{x_i} p_1(x_i) \log \frac{p_1(x_i)}{p_2(x_i)} \, dx_i \\
= \sum_{i=1}^{n} KL(p_1(x_i), p_2(x_i))
\]

(1)

This gives rise to the following representation. We partition the frequency domain into small regions and derive the corresponding spatial filters. We convolve an image using the filters and compute the marginal distributions. Each image is then represented by a vector consisting of all the marginal distributions. We shall call this representation spectral representation.

While this representation can be used directly to characterize rigid objects, objects are often subject to deformations. Here we adopt the learning from examples methodology and use a standard multiple-layer perceptron to learn a classifier for each image type based on this representation.

3. Filter Selection

Given a large number of filters, it can be shown that the marginal distributions are sufficient to represent an arbitrary image up to a translation. Intuitively each filter imposes some constraints on all the images that have the same marginal distributions, and with sufficiently many filters, all the images with the same marginal distributions will be identical up to a translation.

Under this representation, a critical question for image classification and recognition is how to select filters such that the performance is maximized. Our idea is to maximize the performance of the given training data. It is easy to see that histograms are nonlinear operators and this makes an analytical solution for filter selection difficult. Instead, we seek a numerical procedure within the neural network framework. Initially, we train one neural network for each filter and we choose the first to be the one that gives the minimum training error. Then we iteratively choose filters based on the current selection. We choose the next filter as the one that improves the training error most among the remaining filters. The filter selection stops when the error is less than some threshold. The algorithm is summarized in Fig. 2.

This greedy algorithm is computationally efficient but may not guarantee an optimal solution. Because our representation is quite reliable, this procedure works well on all the datasets we have used. As explained in Section 4, the filters selected by our algorithm for each dataset are also consistent with other studies.

4. Results and Comparison

In this section, we demonstrate the effectiveness of our proposed method on three different problems. For all the datasets used here, we start with 40 filters including: (i) Laplacian of Gaussian filters at different scales, and (ii) Gabor filters at different scales with multiple orientations at each scale. We exclude the intensity filter, corresponding to the histogram of the input image, to make our representation illumination invariant. The neural network used here is a standard three-layer perceptron trained using backpropagation. While the number of input units is determined by the filters chosen by the selection algorithm, the hidden units are fixed to be 40.

4.1. 3D Object Recognition

We have applied our method to appearance-based 3D object recognition. For evaluation of our method and comparison with existing methods, we use the Columbia Object Image Library (COIL-100)\textsuperscript{1} dataset, which consists of the

\textsuperscript{1}Available at http://www.cs.columbia.edu/CAVE.

Filter Selection Algorithm

\[
S = \phi \quad B = \{F^{(1)}, \cdots, F^{(K)}\}
\]

repeat

for each filter \( F \in B \)

Train a network with filters \( S \cup \{F\} \)

Compute the training error \( e^F \)

end

\[
F^* = \min_F \{e^F\} \quad e^* = e^{F^*}
\]

\[
S = S \cup \{F^*\} \quad B = B \setminus \{F^*\}
\]

until \( e^* < \epsilon \)

Figure 2. Filter selection algorithm. Here \( B \) is the set of all the candidate filters, \( S \) is the set being chosen, \( W \) prior weights for filters, and \( \epsilon \) a threshold.
Table 1. Recognition results of different methods using the 100 objects in the COIL-100 dataset

<table>
<thead>
<tr>
<th>Methods</th>
<th>training/test views per object</th>
<th>36/36</th>
<th>18/54</th>
<th>8/64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our Method</td>
<td></td>
<td>100.00%</td>
<td>99.50%</td>
<td>96.33%</td>
</tr>
<tr>
<td>SNoW [6]</td>
<td></td>
<td>95.81%</td>
<td>92.32%</td>
<td>81.46%</td>
</tr>
<tr>
<td>SNoW with [6]</td>
<td></td>
<td>96.25%</td>
<td>94.13%</td>
<td>89.23%</td>
</tr>
<tr>
<td>Linear SVM [6]</td>
<td></td>
<td>96.03%</td>
<td>91.30%</td>
<td>84.80%</td>
</tr>
<tr>
<td>Nearest Neighbor [6]</td>
<td></td>
<td>98.50%</td>
<td>87.54%</td>
<td>79.52%</td>
</tr>
</tbody>
</table>

color images of 100 3-D objects with varying pose, texture, shape and size. For each object there are 72 images taken at different view angles with 5° apart. Therefore there are 7,200 color image in the entire dataset.

Recently Pontil and Verri [4] applied Support Vector Machines (SVM) to appearance-based object recognition and their method was tested using a subset of the COIL-100 dataset with half for training and the other half for testing. Yang et al. [6] applied Sparse Network of Winnows (SNoW) to recognition of 3D objects and they used the full set of COIL-100 dataset and compared with SVM methods.

As in [6], we vary the number of training views per object. Given the images in the training set, we apply our filter selection algorithm starting with 40 filters. With the chosen filters, a network is trained and the learned network is then used to recognize the testing images at novel views. The unit with the highest output is taken as the result from the system. Table 1 shows our recognition results using different number of training views along with the results reported in [6]. With eight views for training, our system gives a correct recognition rate of 96.3%. If we allow the correct to be within the closest five, the correct recognition rate is 99.0%. Compared to results in [6], Our method gives the best result under all the test conditions and improves significantly when fewer training views are used. This improvement is essentially because our representation is more meaningful than pixel- and edge-based representations [6].

4.2. Face Recognition

In recent years, the problem of face recognition has been studied extensively in the literature. Our method is very different from most of the current methods for face recognition in that it is a general method for image classification and recognition. However, due to the perceptually meaningful representation, our method is also very effective for face recognition as demonstrated using a face recognition dataset.

Here we use the ORL database of faces [2]. The dataset consists of faces of 40 different subjects with 10 images for each subject. The images were taken at different times with different lighting conditions on a dark background. While only limited side movement and tilt were allowed in this dataset, there was no restriction on facial expression.

Because there are only 10 faces per subject, we use 5 of them as training and 5 for testing. Because some images are more representative for a subject than others, we randomly choose training faces to avoid the potential bias on the performance. Then we repeat the same procedure many times to have a better evaluation. It is interesting to see the filters selected by our algorithm for this dataset. It chose four Gabor filters at largest scales to characterize the global face patterns and two Laplacian of Gaussian filters whose scales are comparable with local facial patterns. Table 2 shows the recognition results for 100 trials. Here we report the average performance, the best and worst among the 100 trials. On average we have achieved over 95% correct recognition rate.

Table 2. Correct recognition rates for the 40 face ORL dataset.

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Average</th>
<th>Best</th>
<th>Worst</th>
</tr>
</thead>
<tbody>
<tr>
<td>First correct</td>
<td>95.4 %</td>
<td>98.5%</td>
<td>90.5%</td>
</tr>
<tr>
<td>First three correct</td>
<td>98.9%</td>
<td>100%</td>
<td>96.0%</td>
</tr>
</tbody>
</table>

4.3. Texture Classification

Without any change, we have also applied our method to the problem of texture classification, which has been studied extensively as a separate topic in computer vision. We argue that texture models should be consistent with perceptual models for objects as they need to be addressed within one generic recognition system; we demonstrate here that our method can be applied equally well to the texture classification problem.

To demonstrate the effectiveness of our approach, we use a dataset consisting of 40 textures [3]. Each texture image is partitioned into non-overlapping patches with size 32 × 32 and then all the obtained patches are divided into a disjoint training and test set. As for 3D object recognition and face recognition, we start with the same 40 filters and apply our filter selection algorithm on the training set. The network trained with the chosen filters is then used to classify the patches in the test set. To avoid a bias due to the choice of the training set, we randomly choose the training set for each texture and run our algorithm many times for a better

2http://www.uk.research.att.com/facedatabase.html
3Available at http://www-dbvs.cs.uni-bonn.de/image/texture.tar.gz.
evaluation. We also change the number of patches in the training set to demonstrate the generalization capability of our representation.

Compared to the filters chosen for COIL-100 and ORL datasets, our algorithm chose filters whose scale is comparable with dominant local texture patterns. Table 3 shows the classification result with 100 trials for each setting. This dataset is very challenging in that some of textures are perceptually similar to other textures in the dataset and some are inhomogeneous with significant variations. With as few as 8 training patches, our method achieves a correct classification rate of 92% on average. With half patches used for training, we achieve an average classification rate over 96%.

Table 3. Classification results for the 40-texture dataset.

<table>
<thead>
<tr>
<th>Test-to-training ratio</th>
<th>Average</th>
<th>Best</th>
<th>Worst</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0</td>
<td>92.07%</td>
<td>94.20%</td>
<td>90.22%</td>
</tr>
<tr>
<td>3.0</td>
<td>94.74%</td>
<td>95.83%</td>
<td>93.07%</td>
</tr>
<tr>
<td>1.9</td>
<td>95.64%</td>
<td>96.73%</td>
<td>94.35%</td>
</tr>
<tr>
<td>1.0</td>
<td>96.36%</td>
<td>97.42%</td>
<td>95.16%</td>
</tr>
</tbody>
</table>

To further demonstrate the effectiveness of our method and compare with existing methods, we apply our method to two datasets that were shown to be very challenging for all the methods included in a recent comprehensive comparative study[5]. Randen and Husoy[5] studied and compared close to 100 different methods for texture classification. For a fair comparison, we apply our method to the same dataset using the original experiment setting. We use a training set to train the neural network with filter selection. Then the learned network is applied to a separate test set. The results from our methods are summarized in Table 4. Because the texture images are perceptually quite similar, an accurate perceptual texture model is needed in order to classify the textures correctly. In addition, two different sets of textures for training and testing make the classification even more difficult. The significant improvement demonstrates the necessity of a perceptually meaningful model for texture classification such as the one proposed here.

Table 4. Classification results of the two datasets used in [5]

<table>
<thead>
<tr>
<th>dataset</th>
<th>First correct</th>
<th>First two correct</th>
<th>Best[5]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig. 11(h)[5]</td>
<td>93.49%</td>
<td>97.75%</td>
<td>67.70%</td>
</tr>
<tr>
<td>Fig. 11(i)[5]</td>
<td>92.96%</td>
<td>98.34%</td>
<td>72.20%</td>
</tr>
</tbody>
</table>

5. Discussion

This paper has presented a spectral representation for classification and recognition of images derived from a generative process. While our model is simple, it is very effective for different problems that have been primarily studied separately in computer vision. The marked improvement of our method over existing ones justifies our principle that *a priori* knowledge should be derived from physical generative processes. Not only our approach is generic as demonstrated through different datasets of real images, the representation also provides other advantages such as illumination, rotation, and scale invariance by choosing proper filters.

Our representation along with the filter selection algorithm provides a unified framework for appearance-based object recognition and image classification. Within this framework, the difference among general object recognition, face recognition, and texture classification is the choice of most effective filters. While filters with large scales are most effective for face recognition as faces are topographically very similar, filters whose scales are comparable with texture elements are most effective for texture classification. Our filter selection algorithm chooses the most effective set of filters in this regard. This may lead to a system that is effective for different types of images, a key requirement for a generic recognition system.

References


---

Available at http://www.ux.his.no/~tranden