Pattern Recognition with Localized Gabor Wavelet Grids

Vinay Kumar B

Subex Azure Limited, Bangalore, India

Sai Sharan D R

Accenture, Bangalore, India

Abstract

Pattern Recognition is an example of advanced object recognition which is influenced by several factors such as shape, reflectance, pose, occlusion and illumination which makes it a difficult task. The choice of the pattern representation is crucial for an effective performance of cognitive tasks such as object recognition and fixation. Extending the existing wavelet analysis techniques, this paper introduces Gabor wavelet grids localized within the region of importance (ROI) for a robust pattern recognition system that is capable of recognizing various image classes. The representation of the patterns is based on a Gabor wavelet transform. The features are extracted as a vector of values using an optimally chosen multiple frequency, multiple orientation 2D symmetrical Gabor wavelet grid that spans the region of importance (ROI). This technique of feature extraction is biologically motivated. The purpose of this study is to demonstrate that localized Gabor wavelet grids with a Multilayer Perceptron classifier or Support Vector Machines can be effectively used for recognition of various types of patterns. The focus is on the development of a single (unified) algorithm that can effectively recognize real world patterns. Recognition of close to 100% is demonstrated for some real world patterns like faces, currency notes, postage stamps and typed characters.

Keywords

pattern recognition, Gabor wavelet, multilayer perceptrons, Gabor wavelet grid, wavelet net

1 Introduction

As a broad subfield of artificial intelligence, machine learning is concerned with the design and development of algorithms and techniques that allow computers to "learn" (Ryszard et al., 1983). At a general level, there are two types of learning: inductive, and deductive. Inductive machine learning methods extract rules and patterns out of massive data sets, whereas deductive learning works on existing facts and knowledge and deduces new knowledge from the old. The major focus of Machine learning research is to extract information from data automatically by computational and statistical methods (Mitchell., 1997); hence, machine learning is not only closely related to data mining and statistics but also to theoretical computer science. Machine learning can be viewed as an attempt to automate parts of the scientific method.

Pattern recognition is a field within the area of machine learning. It can be defined as “the act of taking in raw data and taking an action based on the category of the data” (Bishop., 2007). As such, it is a collection of methods for supervised learning. Pattern recognition aims to classify data (patterns) based on statistical information extracted from the patterns or a priori knowledge. A complete pattern recognition system consists of a sensor that gathers the observations to be classified or described; a feature extraction mechanism that extracts the numeric or symbolic information from the observations; and a classification or description scheme that does the actual job of classifying or describing observations, relying on the extracted features.

Over the past three decades, pattern recognition has become a popular area of research in computer vision and one of the most successful applications of image analysis and understanding. Because of the nature of the problem, not only computer science researchers are interested in it, but neuroscientists and psychologists also.

Pattern recognition in the real world is influenced by several factors such as reflectance, occlusion, aging and other noise induced distortions. The challenge is to develop a single pattern representation and recognition system that can effectively work with various image classes and types. The classification or description scheme is usually based on the availability of a set of patterns that have already been classified or described. This set of patterns is termed the training set and the resulting learning strategy is characterized as supervised learning. Learning can also be unsupervised, in the sense that the system is not given an a priori labeling of patterns, instead it establishes the classes itself based on the statistical regularities of the patterns. Let us concentrate our energies on supervised learning behaviours.

Amidst the exhaustive availability of pattern recognition literature (spanning over 25 years), where do we find our motivation? Although some of the published algorithms have demonstrated excellent recognition results, there are still some open problems. The *primary* focus has always been on recognizing patterns of the same class. Some of the algorithms achieve excellent face recognition whereas some others demonstrate good character recognition. But any real world system will have to deploy a *single* pattern recognition method that can recognize the various patterns that the system interacts with. This paper does not aim at recognizing large databases of similar images, but concentrates on achieving a single recognition system (a *unified* approach) for various classes of real world images like currency notes, stamps, typed characters and faces. Though this unified approach simplifies the task of building a multi-class pattern recognition system, it is important to note that the human visual system actually seems to deploy different neural approaches for the recognition tasks (Biederman and Kalocsai., 1997).

Gabor wavelets and its applications in pattern recognition are not new to researchers. In particular, the Gabor functions have already been well explored in the application of face recognition. Some of the proposed face recognition methods use Gabor functions to extract the features and use down-sampling techniques to reduce the number of features and arrive at the optimal feature set. Both, Independent component analysis (ICA) with genetic algorithms (GA) (Chi et al., 2004) and kernel principal component analysis (KPCA) (Liang et al., 2005) have been proposed for arriving at the augmented Gabor feature vector (AGFV) from the extracted Gabor feature vector. But ICA and PCA have some disadvantages, one of the most apparent being computational cost. If it is excessively expensive or time-consuming to gather measurements, then an optimal trade-off between feature reduction and cost has to be calculated. Typically, we must compute a distance between the feature vector in the probe set and each feature vector in the gallery set, both with hundreds and even thousands of dimensions. Thus, recognition which involves Gabor Feature extraction followed by PCA/ICA down-sampling may be unsuitable for online applications because of its prohibitive cost. For every new face introduced into the training set, the PCA training must be performed again. To avoid this, the usual approach in PCA based systems is to start with a dataset that is huge enough to include most of the variations in facial features. But the initial consideration of a huge dataset may not be suitable for all kinds of applications. In the proposed method (where the patterns are faces), the extracted features are optimal for recognition within small datasets and do not require any further down-sampling. This is true even for the other image classes that I have considered, thus saving significant computational complexity and possibly makes pattern recognition suitable for online applications.

Grid based approaches have been deployed in the context of face recognition (Zhao et al., 2003) with template-matching (Ming et al., 2005) and face bunch graphs (Wiskott et al., 1997). The well known Dynamic Link Matching for face recognition (Wiskott and Malsburg., 1996) introduced the concept of rectangular feature grids. Let us extend this grid-centric approach (the distinction being, localization is within the region of importance) to form a Gabor wavelet net (wavenet) for multi-class feature extraction and in turn generic pattern recognition.

Wavelets are functions that satisfy certain mathematical requirements and are used in representing data or other functions. This idea is not new. Approximation using superposition of functions has existed since the early 1800's, when Joseph Fourier discovered that he could superpose sines and cosines to represent other functions. However, in wavelet analysis, the scale that we use to look at data plays a special role. Wavelet algorithms process data at different scales or resolutions. If we look at a signal with a large "window", we would notice gross features. Similarly, if we look at a signal with a small "window", we would notice small features. This makes wavelets interesting and useful.

The wavelet analysis procedure is to adopt a wavelet prototype function, called an analyzing wavelet or mother wavelet. Temporal analysis is performed with a contracted, high-frequency version of the prototype wavelet, while frequency analysis is performed with a dilated, low-frequency version of the same wavelet. Because the original signal or function can be represented in terms of a wavelet expansion (using coefficients in a linear combination of the wavelet functions), data operations can be performed using just the corresponding wavelet coefficients. And if you further choose the best wavelets adapted to your data, or truncate the coefficients below a threshold, your data is sparsely represented.

It was in 1946 that the first time-frequency wavelets (Gabor wavelets) were introduced by Dennis Gabor *(Gábor Dénes),* a Hungarian physicist, who at that time was researching into communication theory (Gabor., 1946). The representation of images by Gabor wavelets is chosen for its biological relevance and technical properties (Olshausen and Field., 2004). The Gabor wavelets are of similar shape as the receptive fields of simple cells in the primary visual cortex (V1). They are localized in both space and frequency domains and have the shape of plane waves restricted by a Gaussian envelope function. Simple cells in the primary visual cortex have receptive fields which are restricted to small regions of space and are highly structured (Marcelja., 1980). A simple model for the responses of simple cells in the primary visual cortex is the linear receptive field (RF). The response of such a cell can be written as a *correlation* of the input data, i.e. an image $I(x)$, with the modeled RF $p(x)$:

$$a_k(x_0) = \int I(x)\, p_k(x - x_0)\, dx \tag{1.1}$$

Jones and Palmer (1987) suggest modelling the shape of the RFs by 2D Gabor filters. Most of the simple cells can also be combined in pairs (Pollen and Ronner., 1981), one has even symmetry and the other has odd symmetry. The modelled RFs of both the cells are combined in a complex notation with the real part corresponding to the cell with even symmetry and the imaginary part corresponding to the cell with odd symmetry. Thus a biologically motivated filter $p_k(x)$, a plane wave restricted by a Gaussian can be formulated (Daugman., 1988) where $'k'$ is the centre frequency of the filter Thus we use Gabor wavelets for feature extraction because it represents the image based on the way the human vision does. Though the brain uses Gabor and Gaussian like functions, the neural processes, in which these are embedded, are still largely unknown.

2 Design and Implementation

The entire system has been modeled as three interconnected sub models. These can be represented by three blocks which as a whole forms our pattern recognition program. The entire system is as shown in figure 2.1. The entire program has been structured into three modules

- Image acquisition and analysis section
- Feature extraction section
- Neural network classifier

Let us consider each of the above sections and understand the design details.

2.1 Image Acquisition and Analysis Section

The input to this section is the image seeking recognition. The image is pre-processed and registered if required. This pre-processing involves the use of image enhancement techniques. The enhancement is with respect to the quality of the image and not with respect to the rotation or orientation of the images. The registration process is used to match two or more images taken from different sensors or from different viewpoints (Brown 1992). Usually real time systems like target detection, medical and satellite imaging and autonomous navigation require the image registration as an intermediate step. In our case, registration will be necessary only if the dataset contains images from multiple sensors or with large variations in expressions and head rotation.

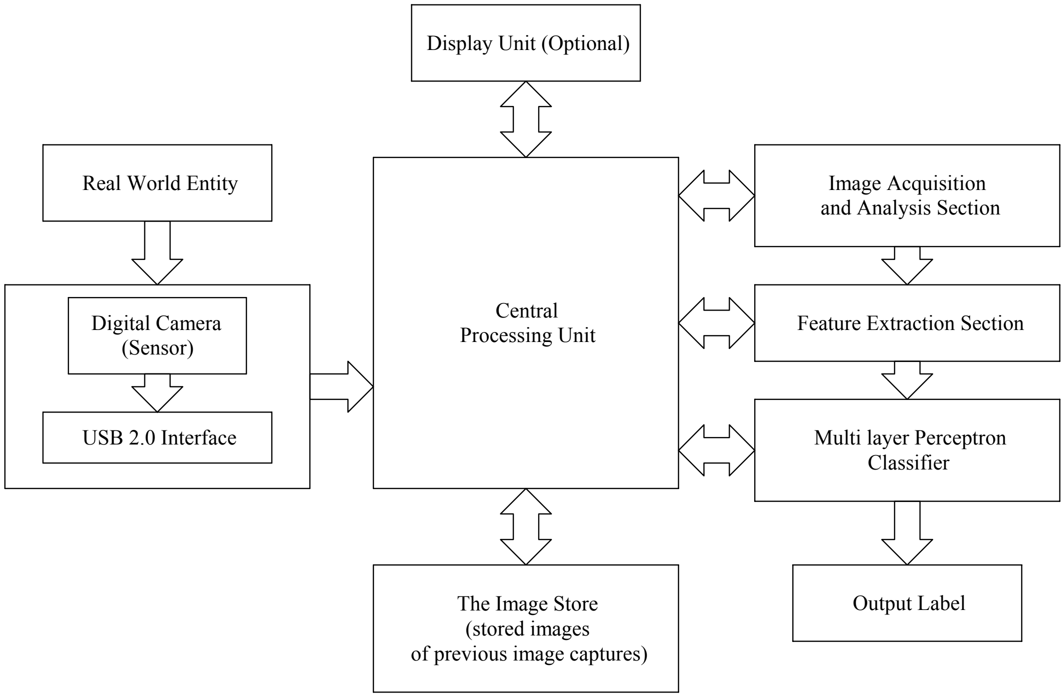

Figure 2.1. Block diagram depicting the various modules in the pattern recognition system.

After the pre-processing step has been completed we need to convert the image to the portable gray map (PGM) format. There are two versions of this format, the `P2’ (ASCII) and `P5' (raw or binary) versions. We utilize the more compact, binary version, i.e., the `P5' format. The advantage of using this standard image format is that images can be easily viewed using almost any of the image display programs. Further, the advantage of using the PGM format is that there exist tools for converting this to and from any other commonly used format.

Once the image has been input to the system, we have to localize the wavelets in the “Region of Importance” (ROI). The ROI varies from image to image. The ROI for stamps or currency notes is the entire image whereas for faces it is the “the inner face region.” Detection of the ROI in the case of faces is not a trivial task. Face detection and feature localization is a separate research subject in itself. The Gabor wavelet grid can be placed within the ROI by manually defining the pixel positions. The better approach is to use automated ROI detection. In the context of face recognition, classic face detection programs, like the one developed at the robotics institute of CMU (Rowley et al., 1998) can be plugged into our image acquisition section for automatic ROI detection. A rectangular grid of feature points is localized within the ROI (manually or automatically detected) as shown in figure 2.3. This wavelet net (wavenet) gives us the uniform distribution of feature points within the ROI. The number of feature points must be chosen optimally. Increasing the number of feature points, gives better representation but at the cost of computational speed. Finding an appropriate trade off between these two factors is of prime importance. The Gabor wavelet expression and feature extraction process is explained in detail in the next section.

We proceed with manual ROI detection. The ROI is different for different image classes. The ROI selection for faces poses challenges compared to the other images and thus is explained in detail. Four points are located on the inner face region as shown in figure 2.2. The red dots indicate the corners of the eyes (represented by points 1 and 2) and the corner of the lips (represented by points 3 and 4).

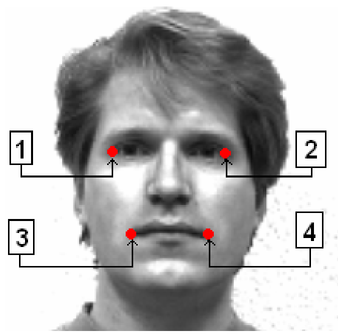

Figure 2.2. Manual detection of facial ROI.

The distance between points 1 and 3 and the distance between points 2 and 4 are calculated. The program calculates the dimensions of the inner region depending upon the ratios of the distances between the above placed points. Now the wavelet net (wavenet) is created and placed in the inner region of the face. The wavenet is then placed on the ROI. A selection of 8x8 points gives the ideal trade off between image representation and computational speed. The wavenet points are as shown below.

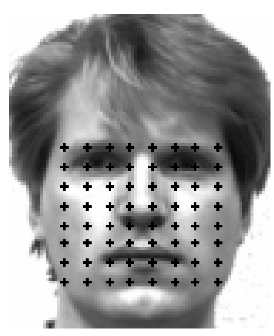

Figure 2.3. The ROI localized 8 X 8 wavelet grid.

2.2 Feature Extraction

A set of frequencies and orientations at each feature point of the wavenet are chosen to represent the image.

Now we define “Lower bound” and “Upper bound” frequencies given by

$$f_{LB} = \frac{1}{x_1\sqrt{2}} \text{ and } f_{UB} = \frac{1}{x_2\sqrt{2}} \tag{2.1}$$

The values of x_1 and x_2 are chosen such that $x_1 > x_2$. A set of frequencies to be used at each feature point is obtained by starting at f_{LB} and multiplying by 2 until f_{UB} is reached. The number of frequencies is given by P. For each frequency, a set of orientations is chosen ranging from $-\pi$ to π. Let Q denote the number of orientations and θ_i denote each of the orientations where

$$\theta_i = \frac{2\pi i}{Q} - \pi \text{ for } 0 < i < Q \tag{2.2}$$

The feature points are denoted by (c_x, c_y) as per the Cartesian co-ordinates. Let the number of feature points be denoted by R. Now we have a set of wavelets with P frequencies and Q orientations at each of the R feature points. Thus, the number of wavelets is

$$N = P * Q * R \tag{2.3}$$

Let $I(x, y)$ or just I represent the input image. Here $'x'$ and $'y'$ represents the coordinates of each pixel in the Cartesian system. Hence $'x'$ and $'y'$ ranges from 0 to $'h'$ and $'w'$ respectively where $'h'$ is the height of the image and $'w'$ is the width of the image.

Now we define a family of N Gabor wavelet functions $\psi = \{\psi_{1,1,1}, \ldots, \psi_{P,Q,R}\}$ of the form

$$\begin{aligned}\psi_{i,j,k}(x, y) &= \frac{f_i^2}{2\pi} \exp\{-0.5 f_i^2[(x - c_{x_k})^2 + (y - c_{y_k})^2]\} \\ &* \sin\{2\pi f_i[(x - c_{x_k})\cos\theta_j + (y - c_{y_k})\sin\theta_j]\}\end{aligned} \tag{2.4}$$

, where f_i denotes the frequency, θ_j denotes orientation and c_{x_k}, c_{y_k} denotes the wavelet position.

Now the wavelet function $\psi_{i,j,k}(x, y)$ is normalized by scaling it by the reciprocal of its length $\|\psi_{i,j,k}(x, y)\|$. Thus we obtain the unit vector $\hat{\psi}_{i,j,k}(x, y)$ as shown below

$$\hat{\psi}_{i,j,k}(x, y) = \frac{\psi_{i,j,k}(x, y)}{\|\psi_{i,j,k}(x, y)\|} \tag{2.5}$$

The first weight w_1 associated with wavelet ψ_1 is given by the dot product of I and $(\hat{\psi}_{i,j,k})_1$

$$w_1 = I \cdot (\hat{\psi}_{i,j,k})_1 \tag{2.6}$$

The subsequent weights w_u for $(2 \le u \le N)$ associated with wavelets ψ_u are determined by calculating the dot products of $I_{(diff)_u}$ and $(\hat{\psi}_{i,j,k})_u$ as shown below

$$w_u = I_{(diff)_u} \cdot (\hat{\psi}_{i,j,k})_u \tag{2.7}$$

, where $I_{(diff)_u}$ represents the intermediate difference image:

$$I_{(diff)_u} = I_{(diff)_{u-1}} - \hat{I}_{u-1} \text{ for } (2 \le u \le N) \tag{2.8}$$

, where $I_{(diff)_1} = I(x, y)$ and $\hat{I}_u$ denotes the intermediate reconstructed image for each wavelet given by

$$\hat{I}_u = w_u \cdot (\hat{\psi}_{i,j,k})_u \text{ for } (2 \le u \le N) \tag{2.9}$$

The intermediate reconstructed images are as shown in figure 2.4 and the final reconstructed image is as shown in figure 2.5

Figure 2.4. The sample intermediate reconstructed images

The final reconstructed image is given by

$$\hat{I} = \sum_{u-0}^{N-1} \hat{I}_u \tag{2.10}$$

Figure 2.5. The final reconstructed image with P = 6, Q = 10, R = 64 (i.e. 8x8)

In the above reconstruction process, a set of $P*Q*R$ weights (projections) are obtained. The weights are scaled between -1 and 1, since the performance of a MLP increases when its inputs are scaled. Then they are sorted on magnitude and the $'n'$ highest weights are selected, where $'n'$ is chosen based on the number of feature points that is required to be fed into the MLP classifier. These $'n'$ weights along with the corresponding set of $'n'$ frequencies and $'n'$ orientations comprise our feature vector. Thus the feature vector input to the MLP comprises of $'3n'$ feature points.

2.3 Neural Network Training and Recognition

Once the features are extracted from the image, they are trained and classified with a multilayer perceptron (MLP). This class of networks consists of multiple layers of computational units, usually interconnected in a feed-forward way. Each neuron in one layer has directed connections to the neurons of the subsequent layer. In many applications the units of these networks apply a sigmoid function as an activation function. The *universal approximation theorem* for neural networks states that every continuous function that maps intervals of real numbers to some output interval of real numbers can be approximated arbitrarily closely by a multilayer perceptron with just one hidden layer. This result holds only for restricted classes of activation functions, e.g. for the sigmoidal functions.

Multi-layer networks use a variety of learning techniques, the most popular being *back-propagation*. Here the output values are compared with the correct answer to compute the value of some predefined error-function. By various techniques the error is then fed back through the network. Using this information, the algorithm adjusts the weights of each connection in order to reduce the value of the error function by some small amount. After repeating this process for a sufficiently large number of training cycles the network will usually converge to some state where the error of the calculations is small. In this case one says that the network has *learned* a certain target function. To adjust weights properly one applies a general method for non-linear optimization task that is called gradient descent. For this, the derivative of the error function with respect to the network weights is calculated and the weights are then changed such that the error decreases (thus going downhill on the surface of the error function). For this reason back-propagation can only be applied on networks with differentiable activation functions.

In general the problem of teaching a network that performs well, even on samples that were not used as training samples, is a quite subtle issue that requires additional techniques. This is especially important for cases where only very limited numbers of training samples are available. The danger is that the network over-fits the training data and fails to capture the true statistical process generating the data. Computational learning theory is concerned with training classifiers on a limited amount of data. In the context of neural networks a simple heuristic, called early stopping, often ensures that the network will generalize well to examples not in the training set.

Other typical problems of the back-propagation algorithm are the speed of convergence and the possibility to end up in a local minimum of the error function. Today there are practical solutions that make back-propagation in multilayer perceptrons the solution of choice for many machine learning tasks.

In this section the feature vector obtained in feature extraction section is processed. The MLP ANN classifier is used for this purpose, since a classifier always attempts to find the closest match. In other words, it never rejects any image but simply outputs the closest match.

The neural network can either be configured in the training mode or recognition mode. Initially the neural network is set to training mode and all the images in the gallery are trained. An optimal output encoding is used to differentiate between individuals. The neural network is trained with respect to all the individuals. Factors such as number of hidden neurons, momentum, learning rate, number of iterations and output encoding are varied for optimal training. While in the recognition mode, the neural network outputs the correct recognition label or the closest match.

3 Test Results

The wavelet net needs to be placed in the region of importance. For manual ROI detection, the placement of wavelets should be consistent with an admissible variation of just few pixels for efficient training and recognition. This problem can be overcome by automatic ROI detection.

Gabor wavelets with 6 frequencies and 10 angles located at each point of an 8x8 wavenet matrix gives optimal trade-off between good representation and computational time required. Thus, we have $P*Q*R$ = 6*10*64 = 3840 projection (weights). Depending on the computational power available the values of *P*, *Q* and *R* can be increased for better representation and recognition.

As mentioned, these obtained weights are scaled, sorted and the highest $'n'$ weights are chosen. The magnitude of the weights was found to be trivial (less than 0.01) for all $n > 50$. Hence, out of the 3840 extracted weights, only the highest 50 are chosen (a simple descending sort on the feature vector). Along with the corresponding 50 frequencies and 50 angles, we have a feature vector with 150 values.

The neural network parameters during training mode were set as follows:

- The number of hidden neurons was set to 20% of the number of input neurons
- Back-propagation training with too small a learning rate will make agonizingly slow progress. Too large a learning rate will proceed much faster, but may simply produce oscillations between relatively poor solutions. Both of these conditions are generally detectable through experimentation and sampling of results after a fixed number of training epochs. Typical values for the learning rate parameter for our datasets is 0.5
- Empirical evidence shows that the momentum of the back-propagation algorithm can be helpful in speeding the convergence and avoiding local minima. The momentum has the following effects:

  a) It smooths the weight changes and suppresses cross-stitching, that is cancels side-to-side oscillations across the error valley.

  b) When all weight changes are all in the same direction the momentum amplifies the learning rate causing a faster convergence.

  c) Enables to escape from small local minima on the error surface.

  The hope is that the momentum will allow a larger learning rate and that this will speed convergence and avoid local minima. On the other hand, a learning rate of 1 with no momentum will be much faster when no problem with local minima or non-convergence is encountered. For a multi class dataset, a momentum of 0.5 gives the optimal trade off between the learning rate and speed of convergence.
- The threshold error level was set to 0.0001, i.e. convergence occurs when the error falls below the threshold

◦ Number of iterations was set to 1,50,001.

For face recognition purposes, the system is extremely robust against facial hair, glasses and small changes in head poise if the number of subjects is less than 10. An efficiency of 97% is observed. But the system is not robust against extreme changes in facial expression. With 50 subjects overall efficiency of close to 95% was achieved.

For recognizing currencies and stamps (50 subjects each), the efficiency is 100%. For typed characters, the efficiency is again 100%. The trained images and test images for the various classes are shown below. The training and testing datasets are distinct.

Class 1: Faces

In order to establish the fact that the algorithm indeed recognizes the various facial variations of the trained subjects, we have considered a diverse testing data-set. The test images constitute of different head poises, changed expressions, addition of glasses, facial hair, gradient and sharp/step changes in illumination and overall decrease in image clarity.

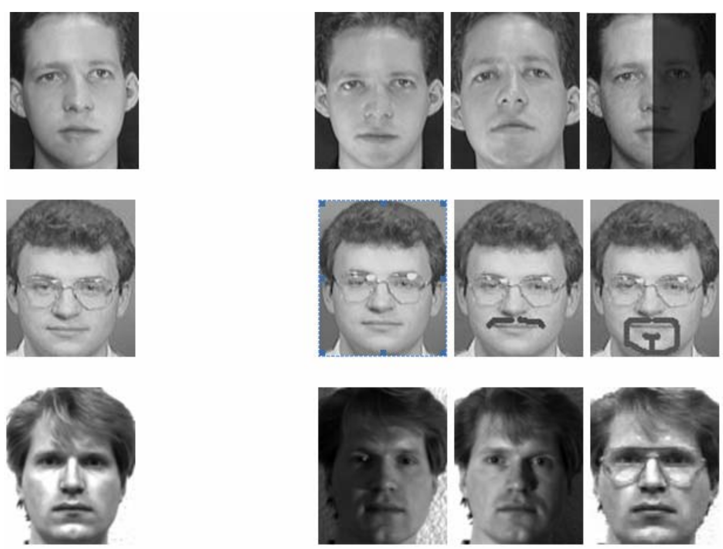

Figure 3.1. The images on the left are the train images and the images on the right are the test images

Class 2: Stamps

The chosen train and test data are diverse; spanning various sizes, images, and patterns. Successful recognition was observed for all the 40 test subjects.

Figure 3.2. The image shows 5 samples out of the 40 postage stamps that were recognized.

Class 3: Typed Characters

Images of all the characters and numerals (A-Z, a-z, 0-9) were successfully recognized. This recognition efficiency is 100% irrespective of the size of the images. The recognition of different fonts is possible with if the neural network is trained to handle the same.

Class 4: Currency Notes

Successful recognition was observed for the 50 test subjects. Some of the test subjects are shown in the image. Clearly visible noise was introduced into the test images to test the robustness of the algorithm. The visible distortions and noise introduced in the image were not a hindering factor was successful recognition.

As you would have noticed, the focus has also been on reducing the mathematical and computational complexity that is usually involved with simulation of a cognitive task like pattern recognition. The feature extraction is complete in precisely ten equations. The MLP takes care of the classification part. To truly appreciate the simplicity, let us compare the system with a PCA system.

Figure 3.3. Some of the sample images. The images on the left are the used for training and the images on the right are the test subjects.

The observations are tabulated in Table 3.1.

Image Class	Number of Subjects	Frequencies (P)	Angles (Q)	Feature Points (R)	Learning Rate	Momentum	Classification Eff. (%)
Faces	15	6	10	64	0.5	0.5	97
Faces	50	6	10	64	0.5	0.5	95
Stamps	50	6	10	64	0.5	0.5	100
Typed Characters	50	6	10	64	0.5	0.5	100
Currency Notes	50	6	10	64	0.5	0.5	100

Table 3.1. Summary of the test results for all the image classes.

3.1 Comparison with PCA

In the PCA approach, each image is treated as a high dimensional feature vector by concatenating the rows of the image together, using each pixel as a single feature. Thus, each image is considered as a sample point in a high-dimensional space. The dimension of the feature vector is usually very large, on the order of several thousands for even small image sizes. The PCA is based on linearly projecting the image space to a lower dimensional space, and maximizing the total scatter across all classes, i.e., across all images of all classes. The orthonormal basis vectors of this resulting low dimensional space are stored. Each image to recognize is then *projected* onto each of these stored basis vectors, giving each of the components of the resulting feature vector. Then, a Euclidian distance is used in order to classify the feature vector.

But, the first drawback of this method is that these vectors that have to be stored, supposes an amount of extra space in the database. A second disadvantage is that images have to be normalized. The third disadvantage is that the PCA method requires quite a long time of training in order to compute the orthonormal basis vectors and the recognition process is as well expensive because it is correlation-based: the test image has to be correlated with each of the basis vectors. Compared to PCA, the proposed method is fast and has a compact storage of the extracted features.

4 Comparison with SVM

For face recognition, the SVM gave better recognition for the same dataset as compared to MLPs. The Gist SVM algorithm (Pavlidis et al., 2004) was used for the identification and verification based on the extracted Gabor features. The support vector machine (SVM) algorithm learns to distinguish between two given classes of data. We train a SVM on a labelled training set and then use the trained SVM to make predictions about the classifications of an unlabeled test set.

A support vector machine is a supervised learning algorithm developed over the past decade by Vapnik, 1995 and others. The SVM algorithm operates by mapping the given training set into a possibly high-dimensional feature space and attempting to locate in that space a plane that separates the positive from the negative examples. Having found such a plane, the SVM can then predict the classification of an unlabeled example by mapping it into the feature space and asking on which side of the separating plane the example lies. Much of the SVM's power comes from its criterion for selecting a separating plane when many candidate planes exist: the SVM chooses the plane that maintains a maximum margin from any point in the training set.

The extracted Gabor features are fed to the set of M SVMs where M is the number of subjects we have. In order to achieve high efficiency, one SVM is trained for each person. The faces will be used as positive examples for their own networks and negative examples for the other networks. Each SVM outputs the recognition label a decision algorithm. The decision algorithm selects the output label based on the signal it receives. The SVMs can be individually used for the open universe recognition task of verification and collectively used for closed universe recognition task of identification. If the decision algorithm receives no positive signal or conflicting signals from two or more SVMs, it does not produce the output label but redirects back to the feature selection process. This further improves the efficiency and reduces probability of error.

4.1 Experimental Results

The SVM parameters during training mode were set as follows:

- The kernel that uses the simple dot product was used, ($K(x_i, x_j) = x_i^T x_j$) making the SVM a linear classifier (the feature space and the input space are equivalent)
- The convergence threshold was set to 1e-06. Training halts when the objective function changes by less than this amount. A larger value will tend to speed execution, at the possible cost of some accuracy.
- The number of iterations limited between 100 to 150 is found to be sufficient.

System is very robust against extreme changes in facial expression if the SVMs have been trained to handle expression changes (i.e. training the SVM for different expressions).

The same dataset that was used for the MLP experiments is used here too. With 15 subjects, a consistent overall efficiency of close to 100% was achieved (it was 97% with MLPs). An increase in efficiency of 3% is observed.

5 Limitations

- In the context of face recognition, the system is not so robust against extreme variations in expression.
- The system cannot be used on skewed or affine transformed images unless explicitly trained to handle such images.

- The computing power that is in general available makes pattern recognition suitable for offline applications as online applications require higher computing power.
- Identical twins cannot be distinguished with our pattern recognition system.

6 Scope for Future Work

- The selection of the ROI is having a significant influence on the recognition. Techniques for automatic detection of ROI “must” be explored and is the next logical extension for the proposed system.
- Image registration can be used if the images are from various sensors and have large variations.
- The system can be made to constantly re-train itself with the latest feature vector that it has successfully recognized. This will make the system constantly adapt to age related changes in the patterns.
- A complete system can be built that will differentiate different classes of patterns and recognize within these classes.
- The system must be made more robust to handle a much larger database.

7 Conclusion

This paper presented a single pattern recognition system using Gabor wavelets localized within the region of importance that could successfully recognize various image/pattern classes. As said earlier, the use of Gabor wavelets is biologically motivated. This approach appears to be quite perspective, insensitive to homogenous illumination changes, robust against facial hair, glasses and also generally very robust for noises in the image as compared to other methods. However it was found to be sensitive to large variations as compared to train images or extreme distortions in the image. Also, it was found that placement of wavelets should be consistent for efficient recognition. Pattern recognition systems are no longer limited to classification, identity verification and surveillance tasks. Growing numbers of applications are starting to use pattern recognition as the initial step towards interpreting human actions, intention, and behavior, as a central part of Next-Generation Smart Environments.

Acknowledgments

I thank Dr. C.N.S. Ganesh Murthy and Dr. Volker Krueger for their support, encouragement and guidance. The images used in the face recognition tests are from the Yale and Olivetti face database.

References

1. I. Biederman, P. Kalocsai. Neurocomputational bases of object and face recognition. *Philosophical Transactions – Royal Society of London Series: Biological Sciences,* 352(1358):1203-1220, 1997.

2. C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.

3. L. G. Brown. A survey of image registration techniques. *ACM Computing Surveys*. 24(4):325-376, 1992

4. W. Chi, G. Dai and L. Zhang. Face recognition based on independent Gabor features and support vector machine. *Fifth World Congress on Intelligent Control and Automation, WCICA*, 5:4030-4033, June 2004.

5. J. G. Daugman. Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech, and Signal Processing,* 36(7):1169-1179, 1988.

6. D. Gabor. Theory of communication. *Journal of Institute of Electrical Engineers*, 93:429-457, 1946

7. J. P. Jones, L. A. Palmer. An evaluation of the two-dimensional Gabor filter model of simple receptive fields in the cat striate cortex. *J. Neurophysiology*, 58:1233-1258, 1987.

8. Y. Liang, et al. Gabor features-based classification using SVM for face recognition. *Lecture Notes in Computer Science*, 3497:118-123, 2005.

9. S. Marcelja. Mathematical description of the responses of simple cortical cells. *J. Opt. Soc. Amer.,* 70(11):1297-1300, 1980.

10. A. Ming, H. Ma, H. Zhang, W. Xiong. A grid-based face recognition approach using general template matching. *First International Conference on Semantics, Knowledge and Grid (SKG'05)*, pages 43, 2005.

11. T. M. Mitchell. *Machine Learning*. McGraw Hill, 1997.

12. B. A. Olshausen, D. J. Field. What is the other 85% of V1 doing? *23 Problems in Systems Neuroscience.* Oxford University Press, 2004.

13. P. Pavlidis, I. Wapinski, W. S. Noble. Support vector machine classification on the web. *Bioinformatics*, 20:586-587, 2004

14. D. A. Pollen, S. F. Ronner. Phase relationships between adjacent simple cells in the visual cortex. *Science,* 212(4501):1409-1411, 1981.H. A. Rowley, S. Baluja, T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23-38, 1998.

15. S. Ryszard, Michalski, J. G. Carbonell, T. M. Mitchell. *Machine Learning: An Artificial Intelligence Approach.* Tioga Publishing Company, 1983.

16. V. N. Vapnik. *The Nature of Statistical Learning Theory*, Springer, 1995.

17. L. Wiskott, J. Fellous, N. Kruger and C. Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775-779, 1997.

18. L. Wiskott and C. Malsburg. *Face Recognition by Dynamic Link Matching*. Lateral Interactions in the Cortex: Structure and Function, chapter 11. The UTCS Neural Networks Research Group, Austin, TX, 1996.

19. W. Zhao, R. Chellappa, A. Rosenfeld and J. Phillips. Face recognition: A literature survey. *ACM Computing Surveys*, 12:399-458, 2003.