Face Recognition using Fractal Dimensions (FD) based Feature

Gayatri Soni¹, Surendra Vishwakarma²
¹M.E. Computer Science, PIT Bhopal, (M.P), INDIA
²Guide, M.E. Computer Science, PIT Bhopal, (M.P), INDIA

ABSTRACT

Face recognition uses are constantly gaining demands because of their requirements human authentication, access controller and surveillance systems. The developers are continuously working to make the system more precise and quicker this paper presents a face recognition system which uses fractal features/component for feature vector formation & then classified using the Support Vector Machine (SVM). Finally the proposed technique is implemented and comprehensively analyzed to test its performance we also compared it with Artificial Neural Network (ANN) classification method.

Keywords-- Fractal Dimensions (FD), Face Recognition, Support Vector Machine (SVM), Artificial Neural Network (ANN)

I. INTRODUCTION

Increasing security demands are forcing the scientists and researchers to develop more advanced security systems one of them is biometric security system. This system is particularly preferred because of its proven natural uniqueness and user has no need to carry additional devices like cards, remote etc. The biometric security systems refers to the identification of humans by their characteristics or traits. Biometrics is used in computer science as a form of identification and access control [1]. It is also used to identify individuals in groups that are under surveillance. One of the biometric identification is done by the face of the person, this method has several application from online (person surveillance) to offline (scanned image identification) etc. The face recognition system has its own advantage over other biometric methods that it can be detected from much more distance without need of special sensors or scanning devices. There are several methods proposed so far for the face recognition system using different feature extraction techniques or different training approaches or different classification approaches to improve the efficiency of the system. The rest of the paper is arrange as the second section presents a short review of the work done so far, the third section presents the details of Fractal and conclusion in sixth and seven sections Dimensions used in the algorithm, the fourth section presents theory behind SVM.

The proposed algorithm is discussed in fifth section followed by experimental analysis respectively.

II. RELATED WORK

This section presents some of the most relevant work recent work presented by other researchers. Ignas Kukenys and Brendan McCane [2] describe a component-based face detector using support vector machine classifiers. They present current results and outline plans for future work required to achieve sufficient speed and accuracy to use SVM classifiers in an online face recognition system. They used a straightforward approach in implementing SVM classifier with a Gaussian kernel that detects eyes in grayscale images, a first step towards a component-based face detector. Details on design of an iterative bootstrapping process are provided, and we show which training parameter values tend to give best results. Jixiong Wang [3] presented a detailed report on using support vector machine and application of different kernel functions (Linear kernel, Polynomial kernel, Radial basis kernel, sigmoid kernel) and multiclass classification methods and parameter optimization. Face recognition in the presence of pose changes remains a largely unsolved problem. Severe pose changes, resulting in dramatically different appearances, are one of the main difficulties and one solution approach is presented by Antony Lam and Christian R. Shelton [4] presented a support vector machine(SVM) based system that learns the relations between corresponding local regions of the face in different poses as well as a simple SVM based system for automatic alignment of faces in differing poses. Jennifer Huang, Volker Blanz and Bernd Heisele [5] presented an approach to pose and illumination invariant face recognition that combines two recent advances in the computer vision field: component-based recognition and 3D
morphable models. In preliminary experiments we show the potential of our approach regarding pose and illumination invariance. A Global versus Component based Approach for Face Recognition with Support Vector Machines is presented by Bernd Heisele, Purdy Ho, Tomaso Poggio [6]. They present a component based method and two global methods for face recognition and evaluate them with respect to robustness against pose changes. In the component system they first locate facial components, extract them and combine them into a single feature vector which is classified by a Support Vector Machine (SVM). The two global systems recognize faces by classifying a single feature vector consisting of the gray values of the whole face image. The component system clearly outperformed both global systems on all tests. Yongmin Li, Shaogang Gong, Jamie Sherrah, Heather Liddell [7] presented face detection across multiple views (frontal view, owing to the significant non-linear variation caused by rotation in depth, self-occlusion and self-shadowing). In their approach the view sphere is separated into several small segments. On each segment, a face detector is constructed. They explicitly estimate the pose of an image regardless of whether or not it is a face. A pose estimator is constructed using Support Vector Regression. The pose information is used to choose the appropriate face detector to determine if it is a face. With this pose-estimation based method, considerable computational efficiency is achieved. Meanwhile, the detection accuracy is also improved since each detector is constructed on a small range of views. Rectangle Features based method is presented by Qiong Wang, Jingyu Yang, and Wankou Yang [8] presents an efficient approach to achieve accurate face detection in still gray level images. Characteristics of intensity and symmetry in eye region are used as robust cues to find possible eye pairs. Three rectangle features are developed to measure the intensity relations and symmetry. According to the eye-pair-like regions which have been found, the corresponding square image patches are considered to be face candidates, and then all the candidates are verified by using SVM. Finally, all the faces in the image are detected.

III. FRACTAL FEATURE EXTRACTION

A fractal dataset is known by its characteristic of being self-similar. This means that the dataset has roughly the same properties for a wide variation in scale or size, i.e., parts of any size of the fractal are similar (exactly or statistically)to the whole fractal.

![Recursive construction of the Sierpinsky triangle](12)

This can further explained as follows. Consider a bounded set $A$ in Euclidean $n$-dimensional space. The set is said to be self-similar when $A$ is the union of $N$, distinct (no overlapping) copies of itself each of which is similar to $A$ scaled down by a ratio $r$. Fractal dimension $D$ of $A$ can be derived from the relation [10].

$$1 = N_r r^D, \text{or } D = \frac{\log(N_r)}{\log(r)}$$

However, natural scenes practically do not exhibit deterministic self-similarity. Instead, they exhibit some statistical self-similarity. Thus, if a scene is scaled down by a ratio $r$ in all $n$ dimensions, then it becomes statistically identical to the original one, so that (1) is satisfied.

Since it is difficult to compute $D$ using (3) directly.

Hence approximate methods are used in this proposal we are using the Differential Box-Counting (DBC) Approach because of its computational simplicity. The working of DBC can be explained as follows: as we have a basic equation of FD presented in equation (1).
Consider that the image of size $M \times M$ pixels has been scaled down to a size $s \times s$ where $\frac{M}{2} \geq s > 1$ and $s$ is an integer. Then we have an estimate of $r = s/M$. Now considering the image as a 3-D space with $(x,y)$ denoting 2-D position and the third coordinate $(z)$ denoting gray level. The $(x,y)$ space is partitioned into grids of size $s \times s$. On each grid there is a column of boxes of size $s \times s \times s$. If the total number of gray levels is $G$ then $s = \frac{M}{G}$. For example, see Fig. (2), where $s = s' = 3$. Assign numbers $1, 2, \ldots$ to the boxes as shown. Let the minimum and maximum gray level of the image in the $(i,j)^{th}$ grid fall in box number $k$ and $l$, respectively. The contribution of $N_r$, in $(i,j)^{th}$ grid.

\[ n_r(i,j) = l - k + 1, \ldots \ldots \ldots (2) \]

For example, in Fig. (3), $n_r(i,j) = 3 - 2 + 1$. (Although in this figure, for simplicity, smooth image surface is taken, but in reality it will be digital image surface.) Because of the differential nature of computing $n_r$, it is named as the differential box-counting (DBC) approach. Taking contributions from all grids, we have

\[ N_r = \sum_{ij} n_r(i,j), \ldots \ldots \ldots \ldots \ldots (3) \]

$N_r$ is counted for different values of $r$, i.e., different values of $s$. Then using (1), we can estimate $D$, the fractal dimension, from the least square linear fit of $\log(N_r)$ against $\log(1/r)$.

Next, we form a two dimensional matrix $FD_{\text{coef}}$ of size $(M \times M) \times d$. That is, $FD_{\text{coef}} = [c_1, c_2, c_3, \ldots, c_{M \times M}]$, where each element $c_{i,j}$ is a vector of size $d$ (total scales) with and each element in is related to the pixel at location $(i,j)$. The final value of $FD_{\text{final}}$ is computed as the fractal slope of the least square linear regression line by [9]:

\[ FD_{\text{final}}(i,j) = \frac{S_{c}(i,j)}{S_{r}}, \ldots \ldots \ldots \ldots \ldots (4) \]

Where $S_r$ and $S_c$ are the sums of squares as follows:

\[ S_r = \sum_{r=1}^{\text{scale}_{\text{total}} - 1} (\log(r))^2 - \left( \frac{\sum_{s=1}^{\text{scale}_{\text{total}} - 1} \log(r)}{\text{scale}_{\text{total}} - 1} \right)^2 \ldots \ldots \ldots \ldots \ldots (5) \]

\[ S_c(i,j) = \sum_{r=1}^{\text{scale}_{\text{total}} - 1} \log(r) \left( c(i,j) \right) \]

\[ \left( \frac{\sum_{r=1}^{\text{scale}_{\text{total}} - 1} \log(r)}{\text{scale}_{\text{total}} - 1} \right)^{-1} \left( \sum_{r=1}^{\text{scale}_{\text{total}} - 1} c(i,j, r) \right) \ldots \ldots \ldots \ldots \ldots (6) \]

The transformation process to convert face images to FD image as illustrated in Figure 4.

Figure 4: showing the Lena image and it Fractal Dimension Image.

IV. SUPPORT VECTOR MACHINE (SVM)

Support Vector Machines (SVMs) have developed from Statistical Learning Theory [6]. They have been widely applied to fields such as character, handwriting digit and text recognition, and more recently to satellite image classification. SVMs, like ANN and other nonparametric classifiers have a reputation for being robust. SVMs function by nonlinearly projecting the training data in the input space to a feature space of higher dimension by use of a kernel function. This results in a linearly separable dataset that can be separated by a linear classifier. This process enables the classification of datasets which are usually nonlinearly separable in the input space. The functions used to project the data from input space to feature space are called kernels (or kernel machines), examples of which include polynomial, Gaussian (more commonly referred to as radial basis functions) and quadratic functions. By their nature SVMs are intrinsically binary classifiers however there are strategies by which they can be adapted to multiclass tasks. But in our case we not need multiclass classification.

4.1 SVM CLASSIFICATION

Let $x_i \in R^m$ be a feature vector or a set of input variables and let $y_i \in \{+1, -1\}$ be a corresponding class label, where $m$ is the dimension of the feature vector. In linearly separable cases a separating hyper-plane satisfies [8].
The optimal separating hyper-plane can now be found by maximizing (10) subject to \( \sum \alpha_i y_i = 0 \) and all Lagrange multipliers are nonnegative. However, in most real world situations classes are not linearly separable and it is not possible to find a linear hyper-plane that would satisfy (7) for all \( i = 1 \ldots n \). In these cases a classification problem can be made linearly separable by using a nonlinear mapping into the feature space where classes are linearly separable. The condition for perfect classification can now be written as

\[
y_i((w, \phi(x_i)) + b) \geq 1, i = 1 \ldots n \ldots (11)
\]

Where \( \phi \) is the mapping into the feature space. Note that the feature mapping may change the dimension of the feature vector. The problem now is how to find a suitable mapping \( \phi \) to the space where classes are linearly separable. It turns out that it is not required to know the mapping explicitly as can be seen by writing (11) in the dual form

\[
y_i\left( \sum_{j=1}^{l} \alpha_j y_j \langle \phi(x_j), \phi(x_i) \rangle \right) + b \geq 1, i = 1 \ldots n \ldots . (12)
\]

and replacing the inner product in (12) with a suitable kernel function \( k(x_j, x_i) = \langle \phi(x_j), \phi(x_i) \rangle \). This form arises from the same procedure as was done in the linearly separable case that is, writing the Lagrangian of (12), solving partial derivatives, and substituting them back into the Lagrangian. Using a kernel trick, we can remove the explicit calculation of the mapping \( \phi \) and need to only solve the Lagrangian (11) in dual form, where the inner product \( \langle x_j, x_i \rangle \) has been transposed with the kernel function in nonlinearly separable cases. In the solution of the Lagrangian, all data points with nonzero (and nonnegative) Lagrange multipliers are called support vectors (SV).

Often the hyper-plane that separates the training data perfectly would be very complex and would not generalize well to external data since data generally includes some noise and outliers. Therefore, we should allow some violation in (7) and (12). This is done with the nonnegative slack variable \( \zeta_i \):

\[
y_i((w, \phi(x_i)) + b) \geq 1 - \zeta_i, i = 1 \ldots n \ldots (13)
\]

The slack variable is adjusted by the regularization constant \( C \), which determines the tradeoff between complexity and the generalization properties of the classifier. This limits the Lagrange multipliers in the dual objective function (10) to the range \( 0 \leq \alpha_i \leq C \). Any function that is

\[
L_d(w, b, \alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \ldots (10)
\]
derived from mappings to the feature space satisfies the conditions for the kernel function.

The choice of a Kernel depends on the problem at hand because it depends on what we are trying to model. The SVM gives the following advantages over neural networks or other AI methods (link for more details http://www.svms.org).

SVM training always finds a global minimum, and their simple geometric interpretation provides fertile ground for further investigation.

Most often Gaussian kernels are used, when the resulted SVM corresponds to an RBF network with Gaussian radial basis functions. As the SVM approach “automatically” solves the network complexity problem, the size of the hidden layer is obtained as the result of the QP procedure. Hidden neurons and support vectors correspond to each other, so the center problems of the RBF network is also solved, as the support vectors serve as the basis function centers.

Classical learning systems like neural networks suffer from their theoretical weakness, e.g. back-propagation usually converges only to locally optimal solutions. Here SVMs can provide a significant improvement. The absence of local minima from the above algorithms marks a major departure from traditional systems such as neural networks.

SVMs have been developed in the reverse order to the development of neural networks (NNs). SVMs evolved from the sound theory to the implementation and experiments, while the NNs followed more heuristic path, from applications and extensive experimentation to the theory.

V. PROPOSED ALGORITHM

The proposed algorithm can be described in following steps.
1. Perform the log transformation on the original image.
2. Compute the $F_{\text{coef}}$ matrix for 10 different scales or $s_{\text{total}} = 11$ as presented in section 3.
3. Now compute the $F_{\text{total}}$ from the $F_{\text{coef}}$ as presented in section 3.
4. Feature vectors are formed by forming the row matrix of coefficients of $F_{\text{total}}$ as follows:
   $$FV(i) = \begin{cases} 
   (F_{\text{total}}(i)), & \text{if } F_{\text{total}}(i) > \text{th} \\
   0, & \text{otherwise}
   \end{cases}$$
5. These vectors are created for all classes of faces.
6. These vectors are used to train the $N^2(N - 1)/2$ (N is the number of classes or different faces) SVM classifiers as we used one against one method.
7. For detection purpose the input image vectors are calculated in same way as during training and then it is applied on each classifier.
8. Finally the decision is made on the basis of majority of class returned by $N^2(N-1)/2$ classifiers.

VI. SIMULATION RESULTS

We used the ORL database for testing of our algorithm. The ORL database contains 40 different faces with 10 samples of each face. The accuracy of the algorithm is tested for different number of faces, samples and vector length.

The comparison of the proposed method with Artificial Neural Network based method shows that the proposed method outperforms the ANN.

<table>
<thead>
<tr>
<th>Face ID</th>
<th>TRP</th>
<th>TNR</th>
<th>FPR</th>
<th>FNR</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9</td>
<td>1</td>
<td>0</td>
<td>0.1</td>
<td>0.99</td>
<td>1</td>
<td>0.9</td>
<td>0.9474</td>
</tr>
<tr>
<td>2</td>
<td>0.7</td>
<td>1</td>
<td>0</td>
<td>0.3</td>
<td>0.97</td>
<td>1</td>
<td>0.7</td>
<td>0.8235</td>
</tr>
<tr>
<td>3</td>
<td>0.9</td>
<td>0.9667</td>
<td>0.0333</td>
<td>0.1</td>
<td>0.96</td>
<td>0.75</td>
<td>0.9</td>
<td>0.8182</td>
</tr>
<tr>
<td>4</td>
<td>0.6</td>
<td>0.9667</td>
<td>0.0333</td>
<td>0.4</td>
<td>0.93</td>
<td>0.6667</td>
<td>0.6</td>
<td>0.6316</td>
</tr>
<tr>
<td>5</td>
<td>0.8</td>
<td>0.9667</td>
<td>0.0333</td>
<td>0.2</td>
<td>0.95</td>
<td>0.7273</td>
<td>0.8</td>
<td>0.7619</td>
</tr>
<tr>
<td>6</td>
<td>0.8</td>
<td>1</td>
<td>0</td>
<td>0.2</td>
<td>0.98</td>
<td>1</td>
<td>0.8</td>
<td>0.8889</td>
</tr>
<tr>
<td>7</td>
<td>0.9</td>
<td>0.9778</td>
<td>0.0222</td>
<td>0.1</td>
<td>0.97</td>
<td>0.8182</td>
<td>0.9</td>
<td>0.8571</td>
</tr>
<tr>
<td>8</td>
<td>0.8</td>
<td>0.9778</td>
<td>0.0222</td>
<td>0.2</td>
<td>0.96</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>9</td>
<td>0.8</td>
<td>0.9778</td>
<td>0.0222</td>
<td>0.2</td>
<td>0.96</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>7u10</td>
<td>0.5</td>
<td>0.9667</td>
<td>0.0333</td>
<td>0.5</td>
<td>0.92</td>
<td>0.625</td>
<td>0.5</td>
<td>0.5556</td>
</tr>
<tr>
<td>Average</td>
<td>0.77</td>
<td>0.98</td>
<td>0.02</td>
<td>0.23</td>
<td>0.959</td>
<td>0.8187</td>
<td>0.77</td>
<td>0.7884</td>
</tr>
</tbody>
</table>
Table 2: Result for Detection of 10 faces while the SVM is trained by 40 Different Faces with 10 Samples of Each One.

<table>
<thead>
<tr>
<th>Face ID</th>
<th>TRP</th>
<th>TNR</th>
<th>FPR</th>
<th>FNR</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.8</td>
<td>1</td>
<td>0</td>
<td>0.2</td>
<td>0.98</td>
<td>1</td>
<td>0.8</td>
<td>0.8889</td>
</tr>
<tr>
<td>2</td>
<td>0.6</td>
<td>1</td>
<td>0</td>
<td>0.4</td>
<td>0.96</td>
<td>1</td>
<td>0.6</td>
<td>0.75</td>
</tr>
<tr>
<td>3</td>
<td>0.9</td>
<td>1</td>
<td>0</td>
<td>0.1</td>
<td>0.99</td>
<td>1</td>
<td>0.9</td>
<td>0.9474</td>
</tr>
<tr>
<td>4</td>
<td>0.8</td>
<td>0.9778</td>
<td>0.0222</td>
<td>0.2</td>
<td>0.96</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>5</td>
<td>0.8</td>
<td>0.9778</td>
<td>0.0222</td>
<td>0.2</td>
<td>0.96</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0.9</td>
<td>1</td>
<td>0</td>
<td>0.1</td>
<td>0.99</td>
<td>1</td>
<td>0.9</td>
<td>0.9474</td>
</tr>
<tr>
<td>8</td>
<td>0.9</td>
<td>1</td>
<td>0</td>
<td>0.1</td>
<td>0.99</td>
<td>1</td>
<td>0.9</td>
<td>0.9474</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0.9778</td>
<td>0.0222</td>
<td>0</td>
<td>0.98</td>
<td>0.8333</td>
<td>1</td>
<td>0.9091</td>
</tr>
<tr>
<td>10</td>
<td>0.9</td>
<td>0.9889</td>
<td>0.0111</td>
<td>0.1</td>
<td>0.98</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>Average</td>
<td>0.86</td>
<td>0.9922</td>
<td>0.0078</td>
<td>0.14</td>
<td>0.979</td>
<td>0.9333</td>
<td>0.86</td>
<td>0.889</td>
</tr>
</tbody>
</table>

Table 3: Result Comparison for Training and Searching Time

<table>
<thead>
<tr>
<th>Total Number of Faces</th>
<th>Training Time (Sec.)</th>
<th>Searching Time (Sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ANN</td>
<td>SVM</td>
</tr>
<tr>
<td>10</td>
<td>0.071711</td>
<td>0.41355</td>
</tr>
<tr>
<td>20</td>
<td>0.095962</td>
<td>1.6526</td>
</tr>
<tr>
<td>40</td>
<td>0.13708</td>
<td>7.2743</td>
</tr>
</tbody>
</table>

VII. CONCLUSION

This paper presents a Fractal Dimension based feature extraction approach in combination with Support Vector Machine (SVM) for the classification phase, the results are also compared with Artificial Neural Network (ANN) classifier. For the detailed analysis wide facial variations has been considered by taking the ORL database for the conducted experiments. The experimental results indicate that the fractal dimensions provides a good method for feature extraction for face recognition system as the results shows the average accuracy up to 97.4%. Considering the comparison between the SVM and NN the results reveals that the SVM method performs better than ANN when compared for recognition accuracy but when compared for training and detection time the neural network out performs the SVM. However the SVM presently tested with only linear kernel function hence the results can be further improved by using different kernel functions such as Quadratic, Polynomial, and RBF etc. we presently leave this work for future implementations.

REFERENCES

[9] Mohammad Reza Faraji and Xiaojun Qi “Face Recognition under Varying Illumination with Logarithmic

