Gender Identification from Facial Images using Local Texture Based Features

Samyak Datta
Department of Computer Science and Engineering
Indian Institute of Technology, Roorkee

Abhishek K. Das
Department of Electrical Engineering
Indian Institute of Technology, Roorkee

***Abstract*—With the advancement of biometric identification systems, there is a need to automate tasks such as gender classification so that personal identification can be done with minimal human intervention. Identifying the gender of a subject from his/her facial image poses an interesting research problem for the Machine Learning and Computer Vision community. This report presents gender classification from facial images by extracting texture-based LBP features. Instance-based (k-Nearest Neighbor), statistical (Support Vector Machines) and connectionist (Neural Networks) learning algorithms have further been applied to the features and a comparitive analysis of the results has been presented.**

***Keywords*—Local Binary Patterns, Gender classification, Support Vector Machines, k-Nearest Neighbors, Neural Networks**

I. Introduction

Gender classification is an important visual task for human beings because many social interactions critically depend on the correct gender perception. As visual surveillance and human-computer interaction technologies evolve, it has become imperative for machines to have an automatic and real time gender identification component. Gender classification systems find use in a wide range of applications demographic data collection, pre-processing step in face recognition systems, biometrics and security surveillance systems and it can also aid in other similar problem domains such as race and ethnicity detection. Hence, gender prediction from facial images has remained an active area of reserach for the Computer Vision and Machine Learning community.

Gender classification is posed as a two-class (binary) classification problem where the input facial image is to be assigned one of the two classes female or male. Similar to all classification tasks, the steps to solve this problem can also be broadly divided into two phases feature extraction and classification. Feature extraction involves the automatic extraction of relevant facial features and expressing the input facial image as a vector of features. The feature selection is in itself a challenging task because it involves a trade-off between two aspects: selecting the optimal subset of features that represent the face in an efficient and robust manner and ensuring that the feature extraction happens in real-time. Post the feature extraction phase, extensive research has been carried out on the various classification frameworks such as Support Vector Machines, Neural Networks or Linear Discrimant Analysis.

In this work, we apply uniform local binary pattern histograms (LBPH) to extract local texture-based features from cropped and aligned frontal facial images belonging to the Nottingham scans database. Our dataset consists of 50 male and 50 female grayscale images with a resoltion of 438 x 538 pixels. We compare the result of running the k-Nearest Neighbor (kNN), Support Vector Machines (SVMs) and Neural Network (NN) classifiers on the extracted features.

The remainder of the report is organized as follows. Section II briefly reviews the current state-of-the-art in gender classification. Section III discusses the feature extraction and classification methods that have been used in this work. A comparative analysis of the results obtained as a consequence of the work done for the current project are listed in Section IV. Section V concludes with the inferences drawn and the scope for further research.

II. Related Work

First attempts of using computer vision based techniques to gender classification started in early 1990s. Since then, many different techniques for solving the problem of gender identification from facial images have been reported in the literature. The proposed techniques differ in (i) the choice of the facial representation, ranging from the use of simple raw pixels to more complex features such as Gabor responses, and in (ii) the design of the classifier, ranging from the use of nearest neighbor and fisher linear discriminant (FLD) classifiers to artificial neural networks, support vector machines and boosting schemes. For instance, Moghaddam and Yang [1] used raw pixels as inputs to SVMs while Baluja and Rowley [2] adopted AdaBoost to combine weak classifiers, constructed using simple pixel comparisons, into single strong classifier. Both systems showed good classification rates. A comparative analysis on gender classification approaches can be found in [3].

Representation of faces in terms of geometrical features is the most common approach for modeling face processing problems. Bruneli and Poggio [4] have used a geometrical feature based method for gender identification. A set of 16 geometrical features is automatically extracted from frontal view images of people without facial hair. This representation has been used to train two competing hyper basis function networks; one for male and one for female.

Colomb [5] has used a 900 x 40 x 900 fully connected back propagation network. Images are scaled to 30 X 30 size images and rotated to position eye and mouth similarly in each image. Output of the hidden unit has been used as input to final neural network (SEXNET) which produces values 1 for male and 0 for female. Tolba [7] has proposed gender

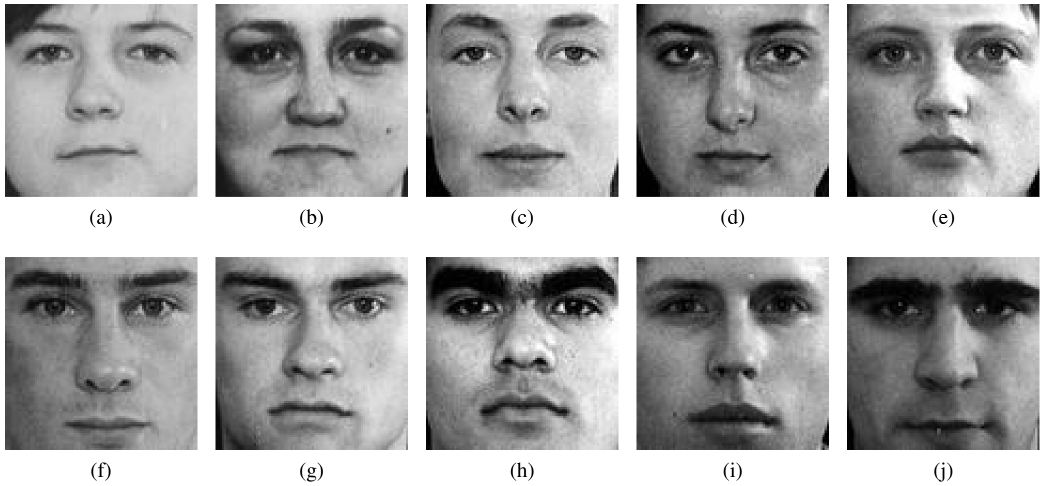

Fig. 1. Cropped and aligned facial images from the Nottingham scans database. (a) - (e) are female whereas (f) - (j) are male faces. All images have a resolution of 90 X 90 pixels.

identification using two different neural network classifiers i.e. radial basis function network (RBF) and learning vector quantization (LVQ). Baback [6] has used SVM classifiers for gender identification from low resolution images (21*12) and has compared its performance on same database with other classifiers.

III. Methods

This section briefly describes the feature extraction and classification methods used in our current work.

A. Local Binary Patterns

Local Binary Patterns (LBP) is a type of feature for texture classification in computer vision. It finds numerous real-world applications because of it's discriminative power, computational simplicity and gray-scale invariance. The original LBP operator labels the pixels of an image by thresholding a 3 x 3 neighborhood of each pixel with the center value and considering the results as a binary number as shown in figure 2. More formally, given a pixel at (x_c, y_c), the resulting LBP can be expressed as:

$$LBP(x_c, y_c) = \sum_{n=0}^{7} s(i_n - i_c).2^n \tag{1}$$

where i_k is the gray-scale intensity value of the pixel with coordinates (x_k, y_k) and $s(x)$ is the function defined as follows:

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & otherwise \end{cases} \tag{2}$$

So, the summation runs over the 8 neighbors (x_n, y_n) of the center pixel (x_c, y_c) with i_c and i_n being the intensity values of the center and the neighboring pixels. These computed binary numbers can be used to represent different local primitives such as corners, curved edges, spots, flat areas, etc as shown in figure 3.

Later, two extensions were made to the original LBP operator. First, the operator was extended to use neighborhood

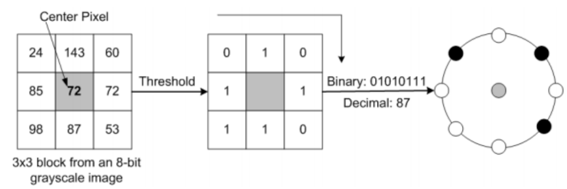

Fig. 2. The working of the basic LBP operator.

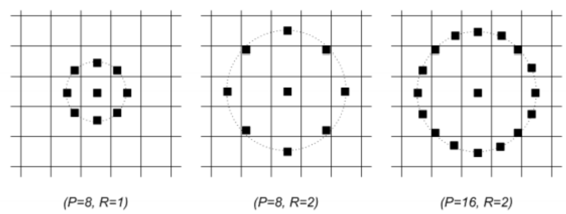

Fig. 3. Example of texture primitives encoded by LBP (black circles represent zeros and white circles represent ones).

of different sizes, to capture dominant features at different scales. The notation $LBP_{P,R}$ denotes a neighborhood of P equally spaced sampling points on a circle of radius of R. Secondly, they proposed to use a small subset of the 2^P patterns, produced by the operator $LBP_{P,R}$, to describe the texture of images. These patterns, called uniform patterns, contain at most two bit-wise transitions from 0 to 1 or vice versa when considered as a circular binary string. For example, 00000000, 001110000 and 11100001 are uniform patterns. It was observed that most of the texture information was contained in the uniform patterns.

After labeling an image with a LBP operator, a histogram of the labeled image can be used as texture descriptor. In the current work, to capture the spatial information, face images are divided into non-overlapping sub-regions; the LBP histograms extracted from sub-regions are concatenated into a single, spatially enhanced feature histogram. The extracted feature histogram describes the local texture and global shape of face images.

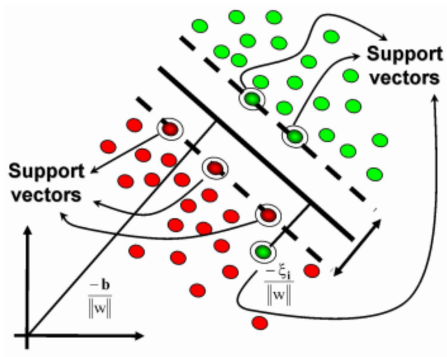

Fig. 4. Binary classification using Support Vector Machines

B. k-Nearest Neighbor Classifiers

k-Nearest Neighbor (or k-NN) algorithm is an instance based (or lazy) learning algorithm where the training examples are simply stored and no description of the target function is explicitly constructed. All computation is postponed until a new instance is to be classified.

The k-NN algorithm assumes that every instance is a point in n-dimensional space and nearest neighbors are defined in terms of Euclidean distance.

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2} \tag{3}$$

where an instance x is defined by $< a_1(x), a_2(x), ..., a_n(x) >$. The target function value (classification result) for a k-NN classifier is the most common value among the k nearest training examples i.e. the target function $f : R^n \to V$ where V is the finite set $\{v_1, v_2, ..., v_s\}$ is given by:

$$\hat{f}(x_q) \leftarrow \arg\max_{v \in V} \sum_{i=1}^{k} \delta(v, f(x_i)) \tag{4}$$

where

$$\delta(a,b) = \begin{cases} 1 & a = b \\ 0 & otherwise \end{cases}$$

Simply put, the k-nearest neighbor classifier assigns the majority class among the k-nearest training points near the query point.

C. Support Vector Machines

Support Vector Machines (SVMs) are primarily binary classifiers that find the best separating hyperplane by maximizing the margins from the support vectors. Support vectors are defined as the data points that lie closest to the decision boundary.

In SVM classification, we are given some training data D, a set of n points of the form:

$$D = \{(\mathbf{x_i}, y_i) \mid \mathbf{x_i} \in R^p, y \in \{-1, 1\}\}_{i=1}^{n}$$

where y_i (either 1 or -1) denotes the class to which the point $\mathbf{x_i}$ belongs to and each $\mathbf{x_i}$ is a p-dimensional real vector. Any hyperplane can be in written in the form of: $w^T x + b = 0$ where w is the normal vector to the hyperplane and the parameter $\frac{b}{||w||}$ denotes the distance of the hyperplane from the origin measured along the normal.

If the training data are linearly separable, we can select two hyperplanes(support vectors) in a way that they separate the data and there are no points between them, and then try to maximize their distance. The region bounded by them is called "the margin". These hyperplanes can be described by the equations:

$$\begin{aligned} w^T x + b &= 1, \\ w^T x + b &= -1, \end{aligned} \tag{5}$$

The distance between these two hyperplanes is $\frac{2}{||w||}$ and hence, the goal is to minimize $|| \ w \ ||$. The SVM classification task is posed as a quadratic optimization (primal) problem as follows: Given the training sample D, find the optimal values of the weight vector w and bias b such that they satisfy the constraints:

$$y_i(w^T x_i + b) \geq 1 \ \forall i = 1, 2, ..., n \tag{6}$$

and the weight vector w minimizes the cost function:

$$\Phi(w) = \frac{1}{2} w^T w = \frac{1}{2} \ || \ w \ ||^2 \tag{7}$$

For the non-linearly separable case, a kernel trick was proposed for the maximum margin hyperplanes. The resulting algorithm is formally similar, except that every dot product is replaced by a nonlinear kernel function. This allows the algorithm to fit the maximum-margin hyperplane in a transformed feature space. The transformation may be nonlinear and the transformed space high dimensional; thus though the classifier is a hyperplane in the high-dimensional feature space, it may be nonlinear in the original input space. Some common kernels are listed below:

1) Polynomial (homogeneous): $k(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i}.\mathbf{x_j})^d$
2) Polynomial (inhomogeneous): $k(\mathbf{x_i}, \mathbf{x_j}) = (1 + \mathbf{x_i}.\mathbf{x_j})^d$
3) Gaussian radial basis function: $k(\mathbf{x_i}, \mathbf{x_j}) = e^{-\gamma||\mathbf{x_i} - \mathbf{x_j}||^2}$

D. Neural Networks

In machine learning and cognitive science, artificial neural networks (ANNs) are a family of statistical learning algorithms inspired by biological neural networks (the central nervous systems of animals, in particular the brain) and are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. Neural networks are typically organized in layers. Layers are made up of a number of interconnected 'nodes' which contain an 'activation function'. Patterns are presented to the network via the 'input layer', which communicates to one or more 'hidden layers' where the actual processing is done via a system of weighted 'connections'. The hidden layers then link to an 'output layer' where the answer is output as shown in figure 5.

Most ANNs contain some form of 'learning rule' which modifies the weights of the connections according to the input patterns that it is presented with. In a sense, ANNs learn by example as do their biological counterparts; a child learns to

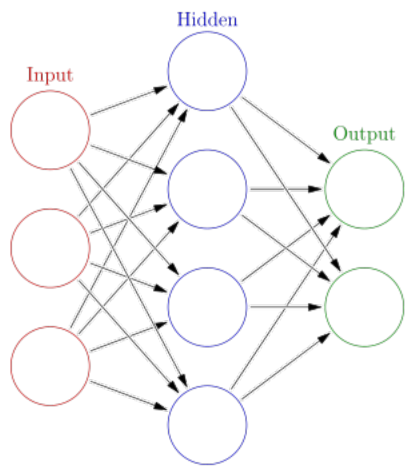

Fig. 5. The architecture of an Artificial Neural N etwork.

recognize dogs from examples of dogs. Although there are many different kinds of learning rules used by neural networks, this demonstration is concerned only with one; the delta rule. The delta rule is often utilized by the most common class of ANNs called 'backpropagational neural networks' (BPNNs).

IV. Experiments

As mentioned in the Introduction, we have used the Nottingham scans database of frontal facial images. For facial analysis tasks such as gender or emotion classification, an accurate alignment of image data is essential for improved results. We have scaled, cropped and rotated the facial images so that they are of the same spatial resolution of 90 x 90 pixels and are aligned with each other. An example of the aligned dataset is shown in figure 1.

After aligning the facial images, we divide each image into 9 sub-regions, each having a 30 x 30 pixel resolution. We then apply the $LBP_{8,1}$ operator to each sub-image separately. While constructing the histogram for each sub-image independently, only the uniform patterns are considered. There are 58 binary codes (betwwen 0 and 255 inclusive) which follow the uniform pattern and each of these 58 grayscale values are assigned their own histogram bins. The remaining 198 grayscale values (the non-uniform patterns) are augmented into a single bin. Hence, using the uniform pattern $LBP_{8,1}$ operator brings down the size of the sub-image LBP histogram from 256 to 59.

All the 9 such histograms are combined in order to form a single spatially enhanced 531-bin global LBP histogram for the image as shown in figure 6. This serves as our feature vector for classification.

For classification, the dataset is split into 80 training and 20 testing examples. k-NN, SVMs and Neural Networks are used for classification and a comparitive analysis is presented in the subsequent section. The optimum value of k for kNN has been selected via 5-fold cross validation. Also, the results obtained by a Bayesian classifier is used as a benchmark.

V. Results and Analysis

Both the feature extraction and classification phases have been studied and implemented as part of this work.

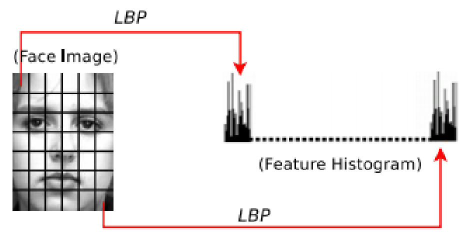

Fig. 6. The facial image with resolution of 90 X 90 pixels is divided into sub-regions from which LBP histograms are extracted and concatenated into 1 single spatially enhanced histogram.

TABLE I. A comparison of the overall accuracy of gender classification systems.

Classifier	Accuracy
Naive Bayes	71
Neural Networks	63
k-Nearest Neighbors ($k = 5$)	73
Linear-SVM	84
Quadratic kernel SVM	85

For texture-based feature extraction, Local Binary Patterns (LBP) has been selected because of it's grayscale and illumination invariance and ease of computation. The result of applying the LBP operator on images are shown in figure 7. Further, various classification algorithms (kNN, SVMs and Neural Networks) have been applied to the features and the results are reported in table I. The naive Bayesian classifier has been used for benchmarking purposes.

The best results are obtained by using an SVM classifier. The hyperparameters of SVM were selected using a 5-fold cross validation technique. The recognition rates of both a linear and a quadratic kernel SVMs are nearly similar with accuracies of 84% and 85% respectively. Table III shows the confusion matrix for the binary classification using a quadratic-kernel SVM.

Table II shows the accuracies obtained by using different values of k for the k-NN classifier. Due to the high dimensionality of the input vector space, k-NN gives decreased performance with the best classification rates of 73% obtained with $k = 5$.

For single hidden layer neural networks the number of hidden units varying from 50 to 100 were tried and 80 and 85 gave the best results. Also, threshold values (a numeric value specifying the threshold for the partial derivatives of the error function as stopping criteria) of 0.003 and 0.005 were tried out.

A comparison between all the classifiers is shown in an illustrative manner in figure8.

VI. Conclusions and Future work

We have demonstrated that uniform local binary pattern histograms are good discriminative features for the task of gender classification from facial images. Comparison has shown that they outperform using raw pixel values as features and are computationally less expensive than more sophisticated techniques such as Gabor filters.

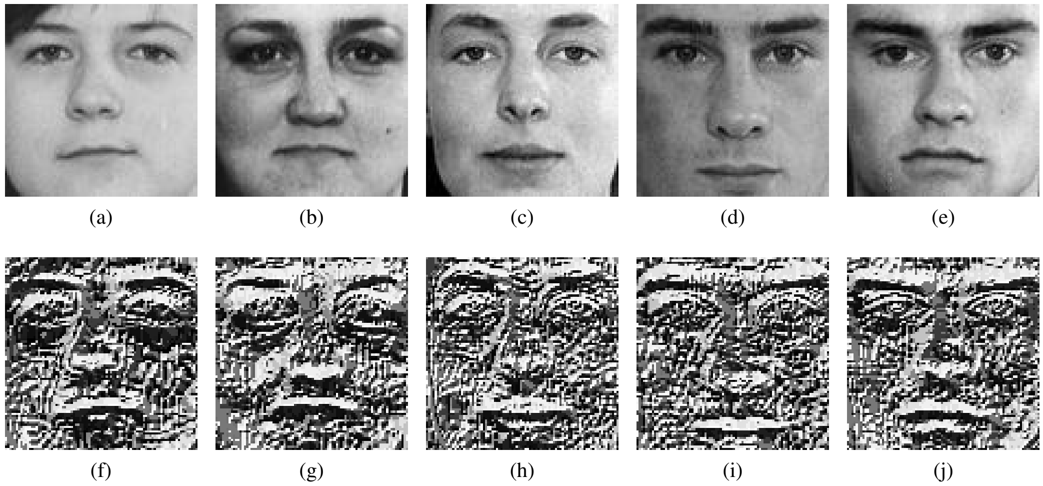

Fig. 7. The ouput on running the LBP operator on the cropped and aligned iages from the Nottingham scans database. (a) - (e) the inpu whereas (f) - (j) are the output images. All images have a resolution of 90 X 90 pixels.

TABLE II. A COMPARISON OF THE ACCURACIES OF K-NN CLASSIFIER FOR DIFFERENT VALUES OF k.

k-Nearest Neighbor	
k	**Accuracy**
3	68
5	73
7	72
9	70

TABLE III. CONFUSION MATRIX FOR CLASSIFICATION USING A QUADRATIC-KERNEL SVM.

		Predicted class	
		Male	Female
Actual Class	Male	82	18
	Female	14	86

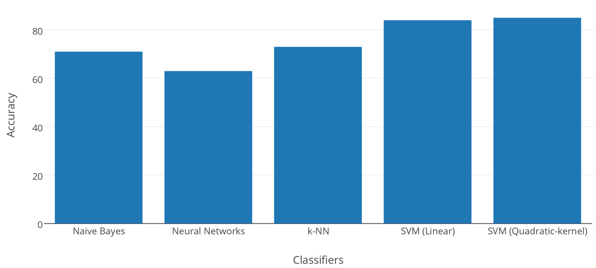

Fig. 8. Comparison between the classification accuracies of different classifiers for gender recognition.

We also compared different types of learning algorithms, namely naive Bayes, kNN, Neural Networks and SVMs. Compared to SVM, both kNN and naive Bayes are very simple and well understood. SVM is however, more appealing theoretically and in practice, its strength is its power to address non-linear classification tasks. However, before we can arrive at any form of general conclusions, the classifiers have to be applied to a wide variety of tasks so that we can validate our claims.

Gender classification opens doors to many interesting applications in allied problem domains. It may be used as a preprocessing step to reduce the search space in face-recognition softwares. More efficient feature extraction techniques can be applied to this problem to enable the classification to happen in real or near real time.

REFERENCES

[1] Moghaddam, B., Yang, M.H.: *Learning gender with support faces*. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5) (2002) 707-711

[2] Baluja, S., Rowley, H.: *Boosting sex identification performance.* International Journal of Computer Vision 71 (2007) 111-119

[3] Makinen, E., Raisamo, R.: *Evaluation of gender classification methods with automatically detected and aligned faces.* IEEE Transactions on Pattern Analysis and Machine Intelligence 30(3) (2008) 541547

[4] R. Brunelli and T. Poggio. *Hyperbf networks for real object recognition.* Proceedings of ECCV, pages 792800, 1992.

[5] B. A. Golomb, D. T. Lawrence, and T. J. Sejnowski. *Sexnet: A neural network identifies sex from human faces.* Advances in Neural Information Processing Systems, 3:572577, 1991

[6] B Moghaddam and M. Yang. *Gender classification with support vector machines.* 4th IEEE international conference on Automatic face and Gesture Recognition, pages 306311, 2000.

[7] A.S. Tolba. *Invariant gender identificaton.* Digital Signal Processing, 11:222240, 2001.