Gender Identification from Facial Images

Deepayan Das, Ashutosh Mishra, Shyam Nandan Rai, Muthireddy Vamsidhar
20172143, 20172087, 20172088, 20172144

January 11, 2018

Abstract

This report is the analysis of applying different methods to the problem of gender classification using facial images. We use PCA to convert images into features which are then used to build different classifiers. We divided the input dataset into training and testing dataset and experiments are performed by varying different parameters relevant to the respective classifier. The outcomes of each of the experiments are analyzed and presented in an apt way.

1 Introduction

Gender classification using facial images has been of interest for quite some time. Humans are very good at determining gender from facial images. Even if the face is cropped to remove all gender cues, we can identify gender with very high accuracy. More recently automated gender classification from facial images has gained much interest in the computer vision and machine learning community. This is because of it's extreme importance in Human Computer Interaction, demographic research, and security and surveillance applications. It can also augment other important areas like face recognition, age and ethnicity determination. Several approaches have been taken to classify facial images based on gender. This report addresses few of these approaches using dimensionality reduction techniques. One of the challenges of automatic gender classification is to account for the effects of pose, illumination and background clutter. Practical systems have to be robust enough to take these issues into consideration. Most of the work in gender classification assumes that the frontal views of faces, which are pre-aligned and free of distracting background clutters, are available. The report is organized as follows - section 2 presents an overview of Methods and dimensionality reduction techniques used. Experiments are illustrated and discussed in section 3.

2 Method

A general pipeline of any image recognition system involves two main parts, Selection of feature type and Selection of classifier type. Both of these are dataset and problem statement specific. In our case, we use the image itself as the feature vector. To remove redundancy in these features, we use Principal Component Analysis(PCA) to reduce their dimension. With respect to classifier type, we experiment with SVM, KNN, Logistic Regression, Naive Bayes, and Bag of Visual Words(BoW). These methods are explained below.

2.1 Preprocessing

Before we featuarize the images, we pre-process them as follows. Each image is resized followed by Histogram Equalization and Intensity Normalization.

2.2 Dataset

We use two datasets in our experiments. The first one, we call this Baseline[Man08] which contains 100 front view images of male and female persons in equal ratio. The second dataset is a miniature version of FaceScrub [KSSMB16] dataset containing 2000 images of male and female persons in equal ratio in different view angles.

2.3 PCA

Principal Components Analysis(PCA) is a well known approach used in reducing dimensionality of the data. The dataset is represented as a matrix $X = [x_1, x_2, \dots x_n]$, where x_i is the i_{th} column vector representing the i_th training image.
The covariance matrix $Q = cov(X) = XX^T$
We then perform eigenvalue decomposition on this matrix X to find the highest ranking eigen vectors by using their eigen values. These eigen vectors are known as principal components and they span the low dimensional sub-space. We choose m eigen vectors$(e_1, e_2, \dots, e_m)$ which best represent the image. The value of m is chosen by considering the cummulative sum of the eigenvalues. The image x is then projected onto the space spanned by these eigen vectors as
$g = [e_1 e_2 \dots e_m]^T x,$
where g is an m dimensional vector. We use this g as a feature during training and classification.

2.4 ICA

Independent Component analysis is a technique to separate out a signal into finer component or into its linear components. It can also separate out the noise/artifacts from the signal as these are independent of the original signal. But, the algorithm lags in recovering the original amplitude of the signal due to whitening.
The algorithm basically starts with the whitening of the data which makes the variance of the data Identity.Then, we rotate the data such that it separates out into its linear components.In, other words the whitened data is rotated such that the gaussianity of the data is minimized.

2.5 Method: SVM

Support vector machines(SVM) are classifiers that construct a maximal separating hyperplane between two classes so that the classification error is minimized. For linearly non-separable data the input is mapped to high-dimensional feature space where they can be separated by a hyperplane. This projection into high-dimensional feature space is efficiently performed by using kernels. For instance-label pair (x_i, y_i) with $x_i \epsilon \boldsymbol{R}^n$ and $y_i \in \{-1, 1\}$ for $1 \leq i \leq n$ where n is the number of instances, the following optimization problem needs to be solved for SVMs –

$$\min_{w,b,\xi} \frac{1}{2} ww^T + C \sum_{i=1}^{n} \xi_i \quad \text{subject to} y_i(\phi(x_i) + b) \leq 1 - \xi \tag{1}$$

In the above equation, C is the penalty parameter for error term and ϕ maps a training instance x_i to higher dimensional space. The kernel K is defined as –

$$K(x_i, x_j) = \phi(x_i)\phi(x_j)$$

2.6 Method: KNN

When the training examples are vectors(x_i) in a multidimensional feature space, each with a class label. The training phase of the KNN-algorithm consists only of storing the feature vectors and class labels of the training samples. In the classification phase, k is a user-defined constant, and a test vector(x) is classified by assigning the label which is most frequent among the k training samples nearest to that test vector(query point). A commonly used distance metric for classification is Euclidean distance

2.7 Method: Logistic Regression

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.

2. Assign each object to the group that has the closest centroid.

3. When all objects have been assigned, recalculate the positions of the K centroids.

4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

Using above written equation, we label a test case x with a class label c_j that achieves the lowest value of J.

2.8 Method: Naive Bayes

Naive Bayes classifiers can handle an arbitrary number of independent variables whether continuous or categorical. Given a set of variables, $X = \{x_1, x_2..., x_d\}$, we want to construct the posterior probability for the event C_j among a set of possible outcomes $C = \{1, c2, c..., cd\}$. In a more familiar language, X is the predictors and C is the set of categorical levels present in the dependent variable. This can be written as

$$p(C_j|X) = p(C_j) \prod_{k=1}^{d} p(x_k|C_j) \tag{2}$$

Using above written Bayes rule, we label a test case X with a class label C_j that achieves the highest posterior probability.

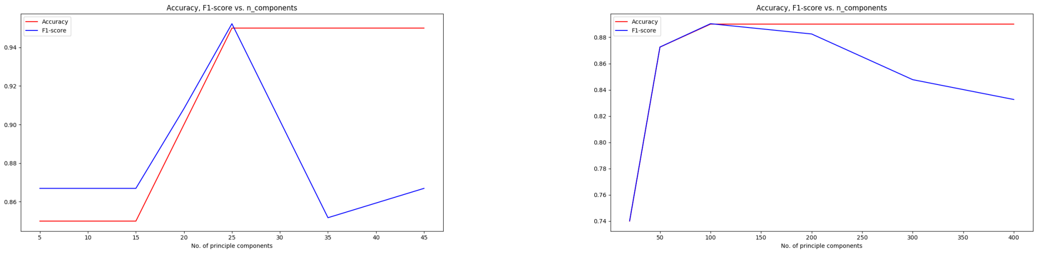
2.9 Method: BoW

In this method, instead of using the complete image as a feature/descriptor, we use non-overlapping patches of the image as features and build a histogram descriptor for the image using them. While selecting patches we select only those which have their variance of Laplacian above a threshold ϵ. After extracting feature vectors from each of the images, we need to construct our vocabulary of visual words. We achieve this using K-means clustering algorithm with clusters(k) which we use to cluster the feature vectors. The resulting cluster centers(c_i) are our dictionary(D) of visual words. Now, we quantize the earlier calculated features based on this dictionary. For each feature vector, we compute its nearest neighbor in the dictionary(D) created using some distance measure like Euclidean distance. We use this set of nearest neighbor labels and build a histogram(h) of length k for each image, where the i_{th} value in the histogram(h) is the frequency of the i_{th} visual word. In short in BoW method, we take a d-dimensional image and represent it using a histogram of k-dimensions. For our case, we then train an SVM on these histograms(h) to classify genders of the images.

3 Experiments and results

All the experiments (except for BoW) were carried out on Base dataset containing 100 images (50 male and 50 female) and on a subset of Face Scrub dataset containing 2000 images (1000 male and 1000 female) by taking the cropped part of the face in the images. Before we extract the features from the image, we pre-process them as follows. Each image is resized followed by Histogram Equalization and Intensity Normalization.

3.1 SVM



(a) (b)

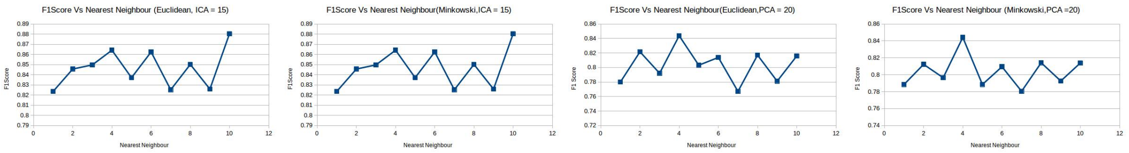
Figure 1: Results of Gender classification on Face images using SVM on validation datasets
ICA:(a) Base Dataset (b) Face Scrub Dataset

3.2 KNN

For the base experiment, the division was the same as mentioned above. We show the results using two distance metrics 'Euclidean' and second 'Minkowski' using the PCA and ICA as the dimensionality reduction techniques. We have calculated the mean f1 score and test accuracy run for 10 splits for 10 nearest neighbors and averaged them.

Base Dataset: On the graph, we have ploted for the component for which we got the highest f1 score and its value for increasing neighbors

- **Minkowski Metric with PCA 20 components vs Euclidean Metric with PCA 20 components**: As can be observed, as the number of nearest neighbor increases, there is an increase in the the accuracy though there are some dips but after NN 8, the accuracy starts to decreases slightly after taking the dip.
- **Minkowski Metric with ICA 15 components vs Euclidean Metric with ICA 15 components**: As can be observed, as the number of nearest neighbor increases, there is an increase in the the accuracy though there are some dips but after NN 8, the accuracy starts to decreases slightly after taking the dip.

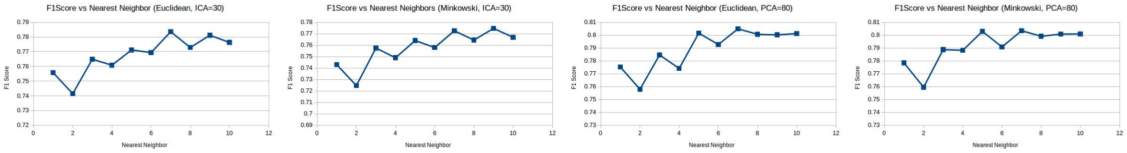


(a) (b) (c) (d)

Figure 2: Results of Gender classification on Face images using KNN on Base testing dataset
ICA:(a) Accuracy-Euclidean distance (b) Accuracy-Minkowski distance
PCA:(c) Accuracy-Euclidean distance (d) Accuracy-Minkowski distance

FaceScrub: On the graph, we have ploted for the component for which we got the highest f1 score and its value for increasing neighbors

- **Minkowski Metric with PCA 80 components vs Euclidean Metric with PCA 80 components**: As can be observed, as the number of nearest neighbor increases, there is an increase in the the accuracy though there are some dips but after NN 7, the accuracy starts to decreases.
- **Minkowski Metric with ICA 30 components vs Euclidean Metric with ICA 30 components**: As can be observed, as the number of nearest neighbor increases, there is an increase in the the accuracy and after NN 7 in Minkowski and 9 in Euclidean, the accuracy starts to decrease.



(a) (b) (c) (d)

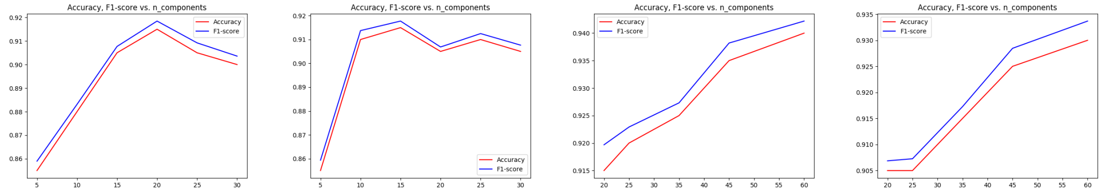
Figure 3: Results of Gender classification on Face images using KNN on Face Scrub testing dataset
ICA:(a) Accuracy-Euclidean distance (b) Accuracy-Minkowski distance
PCA:(c) Accuracy-Euclidean distance (d) Accuracy-Minkowski distance

3.3 Logistic Regression

For both PICS and Face Scrub dataset, we have divided the dataset into 80:20 ratio of training and testing. Subsequently, we apply PCA and ICA on the dataset with varying components for both of them. The choice of the regularization parameter was searched using GridSearchCV function of sklearn. We have calculated the mean f1 score and test accuracy run for 10 splits. The plots for the respective schemas are:

1. L1 Regularization with ICA, F1 Score and Accuracy with increasing number of components.
2. L2 Regularization with ICA, F1 Score and Accuracy with increasing number of components.
3. L1 Regularization with PCA, F1 Score and Accuracy with increasing number of components.
4. L2 Regularization with PCA, F1 Score and Accuracy with increasing number of components.

Base Dataset: From the graph plots, it can be observed that Logistic Regression with L2 regularization and L1 performed equally better with L1 showiung slightly general trend in the case of PCA. For ICA, L1 and L2, both of them gave good results, where L2 gives good accuracy with 15 components and L1 when components were 20.



(a) (b) (c) (d)

Figure 4: Results of Gender classification on Face images using Logistic Regression on PICS test dataset using ICA and PCA
ICA: (a) L1-Accuracy (b) L2-Accuracy
PCA: (c) L1-Accuracy (d) L2-Accuracy

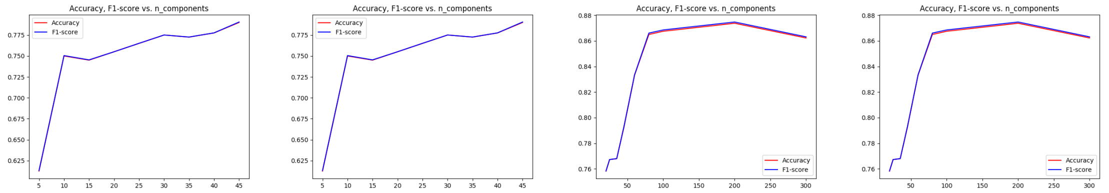
FaceScrub Dataset: In case of PCA, L1 and L2 both acheive the best result of approximately 87% around 200 and 100 components taken into consideration respectively. But for the general trend, L2 performs better as compared to L1.

In the case of ICA, both show plots show similar trend, the accuracy increasing with the increasing number of components with a slight drop when 15 components were taken.

3.4 Naive Bayes

For the base experiment, we divided the dataset into the split explained above and performed GaussianNB using sklearn function. For the two datasets we got the following plots and predictions:

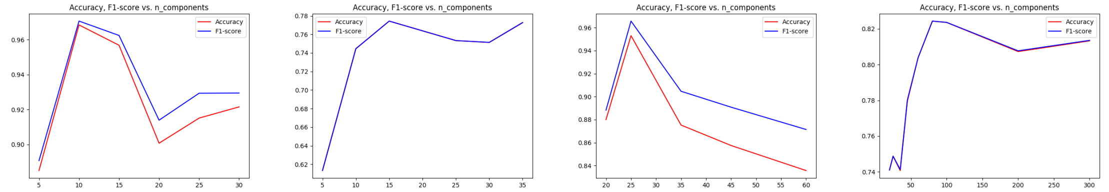
Base Dataset: For the base dataset, the accuracy shows an abrupt behavior as it seeks to increases. In the case of ICA and then suddenly decrease to become constant and in the case of PCA it keeps on decreasing.



(a) (b) (c) (d)

Figure 5: Results of Gender classification on Face images using Logistic Regression on Face Scrub validation dataset using ICA and PCA
ICA: (a) L1-Accuracy (b) L2-Accuracy
PCA: (c) L1-Accuracy (d) L2-Accuracy

FaceScrub: In the case of PCA, it can be seen that the accuracy increases, takes a dip and then slightly increases.In the case of ICA the accuracy increases, takes a dip and then increases after that same as in PCA's case.

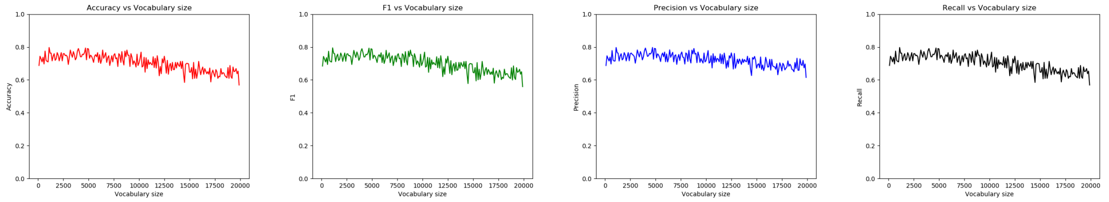


(a) (b) (c) (d)

Figure 6: Results of Gender classification on Face images using Naive Bayes on validation dataset
ICA:(a) Accuracy-base dataset (b) Accuracy-Face scrub dataset
PCA:(c) Accuracy-base dataset (d) Accuracy-Face scrub dataset

3.5 BoW

We divide the Face Scrub dataset into Train, Validation and Test sets. For each set,we take 10*10 patches in the image and take them as features if their variance of laplacian is above a threshold value. We then vary the vocabulary size and test the classifier on Validation dataset and the results are shown in the below graphs. The vocabulary size which gives the best accuracy is chosen and is used to report the results on test dataset.



(a) (b) (c) (d)

Figure 7: Results of Gender classification on Face images using BoW method on validation dataset
(a) Accuracy (b) F1 score (a) Precision (b) Recall

As can be seen in the above figures, the accuracy increases as we start increasing the vocabulary size initially. Then it fluctuates around the same area for a while as we increase the vocabulary size and starts decreasing as we further increase the vocabulary size. It is evident that as we get vocabulary size closer to original number of features, it shows a decrease in the accuracy. We get maximum accuracy on validation dataset at vocabulary size of 1100.

Table 1: Accuracy and F1-score for different methods

	Face scrub		PICS	
Method	Accuracy	F-Score	Accuracy	F-score
PCA+SVM	0.88	0.89	0.95	0.95
BoW	0.73	0.73	-	-
LR + PCA	0.87	0.87	0.93	0.94
KNN + PCA	0.81	0.81	ICA 0.87	0.87
Naive Bayes + PCA	0.82	0.82	0.95	0.96

3.6 Cartoon Space and Cross Modality

Research in gender recognition has been significantly advanced in last few years. Gender databases have triggered the research and study in gender recognition domain. On the other hand, there have been only very few attempts to address the problem of recognizing gender in cartoon faces .Gender recognition is the one of the step towards larger understanding of cartoon images. The problem of gender recognition in cartoon face is closely related to real face gender recognition, however poses many additional challenges. For example,

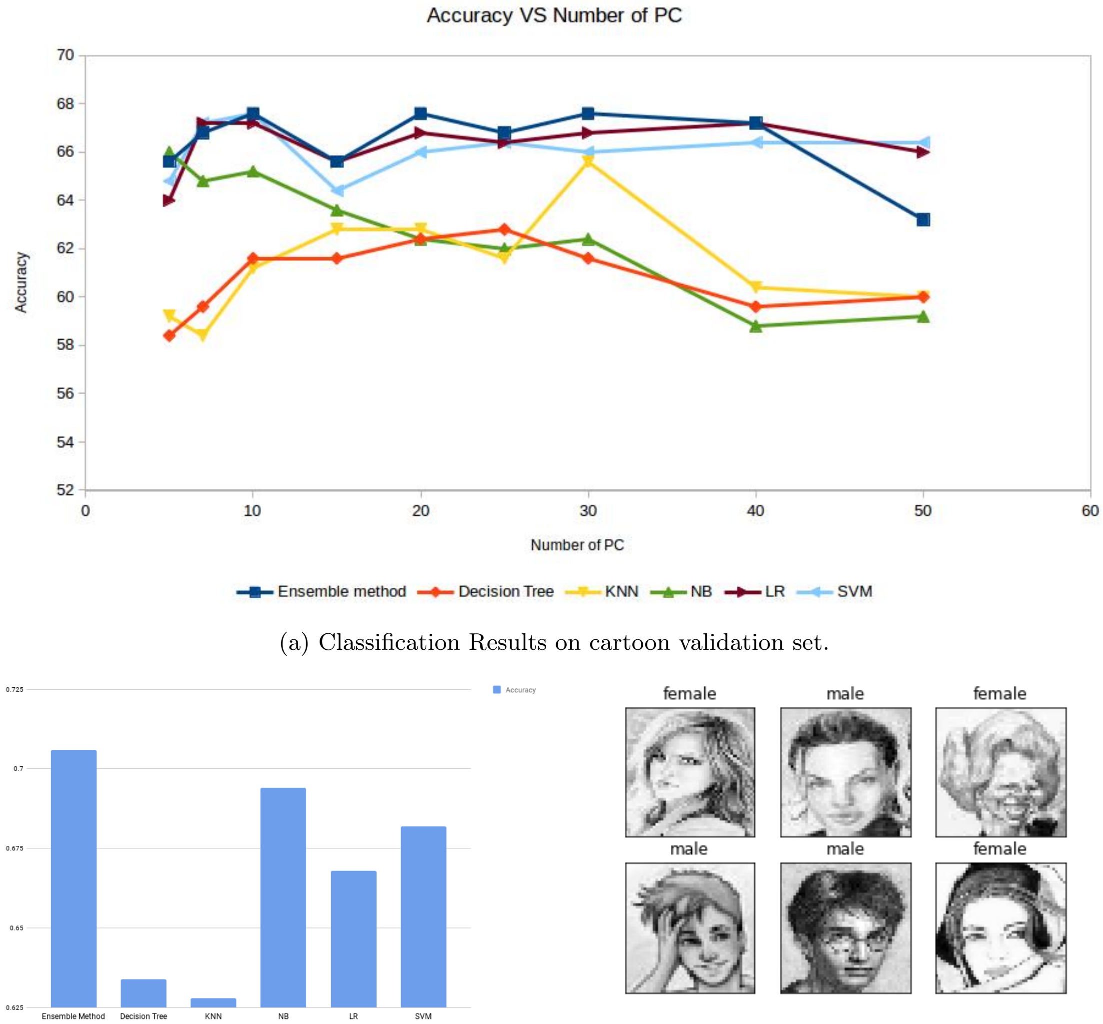
1. artistic variations
2. limited examples
3. magnitude less data than realfaces
4. highly caricatured

The other challenges such as pose,expression, age,illumination variations.

In order to address this problem of identifying gender using facial features in cartoon images [MRMJ16], we have implemented an ensemble of following classifiers:

1. SVM
2. Decision Tree
3. K Nearest Neighbor
4. Naive Bayes
5. Logistic Regression

Intuition: Behind observing the cross modalities between the cartoon faces and real faces is to find whether the gender features found in both the spaces are transferable from one to another i.e. to investigate that whether their exist a correlation between both the spaces.

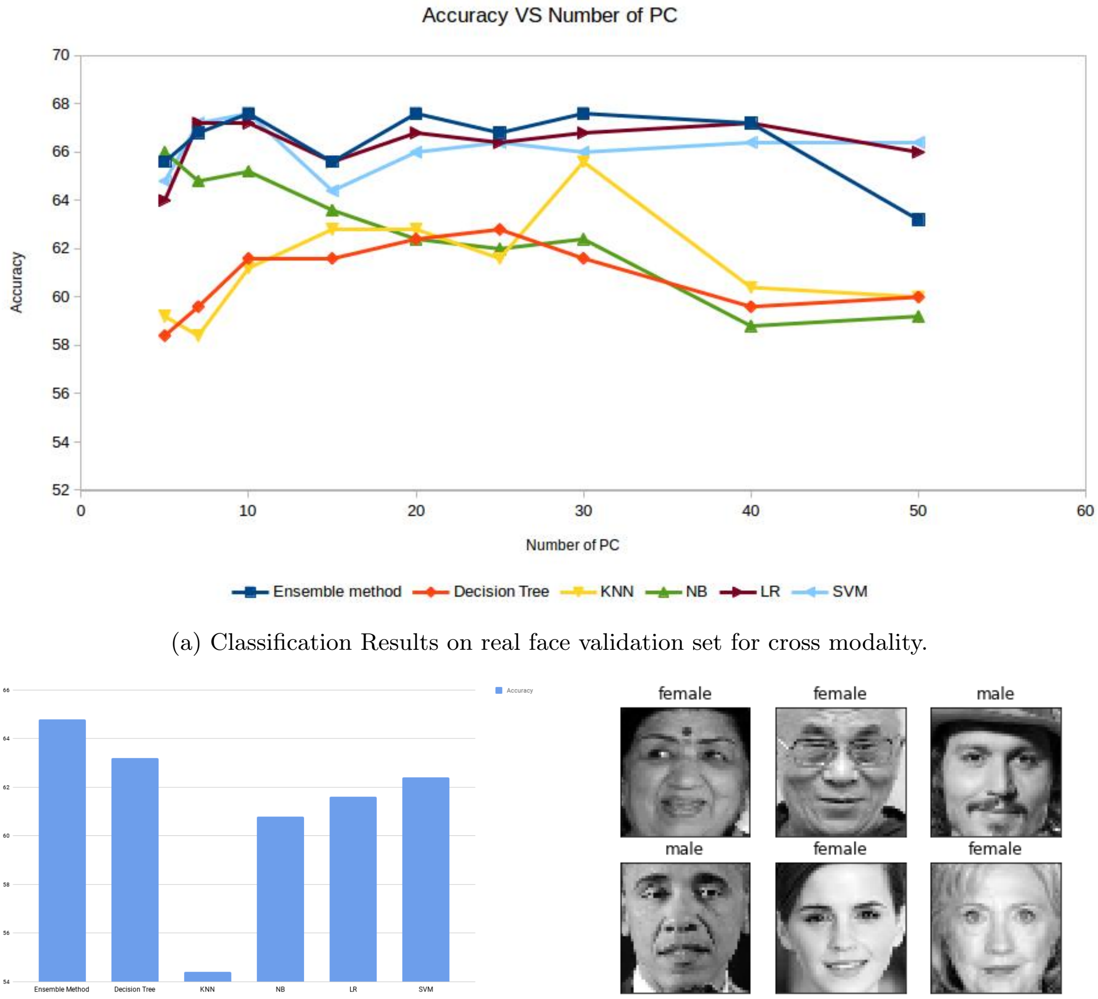


(a) Classification Results on cartoon validation set.

(b) Performance of various classifer on test dataset

(c) Prediction of gender on cartoon images

Figure 8: Results of Gender classification on cartoon images using various set of classifiers on IIITCFW dataset



(a) Classification Results on real face validation set for cross modality.

(b) Performance of various classifer on test dataset for cross modality

(c) Prediction of gender on real images

Figure 9: Results of Gender classification on real images using various set of classifiers trained on IIITCFW dataset.

4 References

References

[KSSMB16] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4873–4882, 2016.

[Man08] Fahim Mannan. Classification of face images based on gender using dimensionality reduction techniques and svm. *School of Computer Science McGill University*, 2008.

[MRMJ16] Ashutosh Mishra, Shyam Nandan Rai, Anand Mishra, and CV Jawahar. Iiit-cfw: A benchmark database of cartoon faces in the wild. In *European Conference on Computer Vision*, pages 35–47. Springer, 2016.