American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)

ISSN (Print) 2313-4410, ISSN (Online) 2313-4402

http://asrjetsjournal.org/

Hair Color Classification in Face Recognition using Machine Learning Algorithms

Saman Sarraf*

Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, Canada, The Institute of Electrical and Electronics Engineers, IEEE

Email: samansarraf@ieee.org

Abstract

Security through automatic human identification is critically important today, and this is largely due to the high volume of communications. Most methods used to identify individuals often use biometrics information, such as facial characteristics. Therefore, face recognition and classification have garnered great interest among computer vision researchers over the past decade. This pattern recognition problem is divided into several subcategories, such as eye or hair detection and classification. Hair is a salient feature in the human face and is one of the most important cues in face detection and recognition. Accurate detection and presentation of the hair region is one of the key components in the automatic synthesis of human facial caricature. In this work, hair color classification through feature extraction and machine learning methods was performed. The impacts of different features and classifiers were investigated using color samples. Support vector machines (SVM) and Kth nearest neighbors (K-NN) were trained by variety sets of statistical and color features, and the trained models were validated. Additionally, the effects of the size of datasets and feature dimensionality reduction were obtained. The best accuracy rate of 99% was achieved through a support vector machine with radial basis kernel function (SVM-RBF) using nine selected statistical and color features.

Keywords: Face Recognition; Hair Color Classification; Machine Learning.

1. Introduction

Face recognition is one of the most important concepts in biometrics, and it can be considered in the research or marketing of surveillance software.

----

* Corresponding author.

From camera surveillance to cosmetic surgery, its applications are widely visible. Today, many researchers strive to recognize differences in facial characteristics by using mathematical algorithms with the help of several facial features, such as the eyes, lips, etc. One feature that has not been examined as closely as other features is hair. The variety of hair models and colors would render such an examination interesting, but there is always the question of whether hair is a good feature to be incorporated in face recognition. This course project will concentrate on a binary classifier that can recognize black hair from non-black hair with the use of SVM and K-NN classifiers. Although there are some robust characteristics to detect faces, hair and its color demonstrate significant potential for study. As is well-known, SVM is a powerful classifier, especially when a binary decision is needed, and K-NN is a rapid approach to data clustering. Neeraj Kumar and his colleagues demonstrated in several projects that different facial features can be useful in search engines when there are large collections of images, including facial images [1]. In fact, their works demonstrate how almost all features of the face are important to consider. For instance, between the studied attributes, sunglasses returned better results than eyeglasses, etc [2]. Generally speaking, the method used by Neeraj and his colleagues was based on low-level feature extraction using different powerful classifiers, such as SMV and Adaboost. As mentioned before, SVM and K-NN were used here because of their advantages, which are described in upcoming chapters. In this work, the classifiers were trained by a set of 460 black and non-black hair images and tested by 200 images. As a short-term project, this number of images would be acceptable to train a classifier, but for a complete project much more images, potentially thousands, are required to train classifiers. The more images used in training, the more reliable the trained classifier. Different parameters derived from feature extraction and the numbers of images used for training have been compared in order to demonstrate how changes in parameters can improve the results, and vice versa. Acceptable results have been obtained, and this will also be discussed in upcoming chapters. Training SVM as well as K-NN classifiers, studying the effects of features on the results by computing recall, and producing precision and accuracy graphs are the principal objectives of this report. The main conclusion of this report is that the choice of a classifier and (low-level) feature extraction is important when black hair is the only feature involved in color image processing. Some suggestions for improving the results will be presented in the final chapter.

2. Background

The concept of hair recognition is derived from the features considered in face detection. Over the last several years, many algorithms have been developed in order to obtain the highest level of accuracy in this field of research. As is well-known, the hair region that comprised a portion of the face is a unique characteristic that can distinguish two people with similar facial structures. This feature [hair] has an effect on facial image analysis used in human identification, for instance, in gender classification [3]. The most significant characteristic of hair that must be considered is its color. Most methods used in hair color detection are based on colors, and with the help of two color spaces, RGB and HSV, acceptable results can be achieved. HSV is a color space dividing the image in three values: hue, saturation and color, respectively. The advantage of using HSV is to remove the impact of luminosity during the process. Information related to brightness can be found in the V value [4]. On the other hand, some other characteristics could be studied for a hair detection project, such as volume and symmetry [3]. However, in this work, the main feature chosen is color, because the binary classifier must distinguish black hair from other hair colors. Two prevalent algorithms used for hair and hair color detection are based on color [1] and region selection [3]. Unfortunately, not many papers were found on the Internet or at the Concordia University library about hair color and region selection, which could indicate the possibility of additional work in this field and the publication of more papers. In the color based approach (which will be explained more thoroughly in upcoming chapters), RGB and HSV values are considered in feature extraction, but in the region-based algorithm, a geometrical model for hair is proposed. Following the feature extraction step, it is possible to use a classifier in most cases. The most useful classifier that has been mentioned in the extant literature is the Super Vector Machine[1,2,5]. Some other classifiers, such as the Artificial Neural Network [6,7] and the K^{th} Nearest Neighborhood [5], have been used as well. The focus of this experience in classification is on the Super Vector Machine, or the SVM. This tool is a statistical method that has been identified as one of the strongest tools in pattern recognition. As is well-known, this supervised learning method is especially used for classification, and the most important characteristic that makes it very famous is its strong reputation in the field of handwriting recognition. The main idea belongs to “Vapnik 1998,” which is represented as “apply a linear method to the data but in a high dimensional feature space” [8]. “Radial Basis Function called RBF” is a popular kernel function [9] described by:

$$K(x,x') = exp\,(-\gamma \lVert x - x' \rVert^2)$$

Equation 1

$f(x) = \sum_{i=1}^{m} \alpha_i \exp(-\gamma \lVert x - x' \rVert^2)+b$

Equation 2

In order to compare the result of SVM with other classifiers, the K^{th} Nearest Neighborhood has been used. K-NN is one of the simplest classifiers in pattern recognition that calculates local approximations. As is well-known, the primary aim of clustering algorithms is to separate the given data between different groups by combining their similarities. K-NN was presented for the first time by E. Fix and J. Hodges in 1951, and this method is based on distance (Euclidean or Mahalanobis) calculations between all points in a dataset.

$$d = \sqrt{\sum_{i=0}^{n} (x_i - y_i)^2}$$

Equation 3

Neeraj and his colleagues used the recall-precision curve and accuracy in order to evaluate the results obtained from their algorithms. Another approach called Labeled Face Wild (LFW) was also used, but in this project, the first approach was employed.

3. Theory and Methodology

The implementation of black hair recognition based on a classification theory was conducted in several steps, as explained below:

3.1. Data Collection

The first step of the project is data collection, which took a long time because it was done manually. One set of data was selected in the personal archive. The second portion of data collection was obtained by surfing the Internet, especially "Google," and the last set of images came from a database called "Face detection dataset and benchmark". The major issue with this work was to find frontal images, such as passport-size photos with black and other hair colors. Another small problem involved distinguishing between which image really features black hair and which image features another hair color that was created as a result of a personal decision. Finally, more than 800 images were downloaded and studied, and a dataset of 660 images including 329 black hair and 331 other color images obtained.

The cropping process was then conducted using "Adobe PhotoShop CS3" software without any other significant manipulation. This process was based on cropping images to a complete frontal facial image. Images were geometrically aligned in a manner that allowed all eyes to be lined up horizontally. The cropped images did not have the same size, so size normalization was required. By an implemented code in MATLAB, all images were reshaped to an equal size. Size normalization did not change facial shapes or result in pixelization.

3.2. Feature Extraction

The most important aspect of this project was how to choose features and set up feature extractions. The most prevalent pattern recognition algorithms in image processing are designed "by extracting low-level features in images," such as pixel values and intensity, gradient etc. [2]. As mentioned previously, features related to two color spaces were selected, and this decision was made based on three papers produced by Neeraj Kumar and his colleagues [2, 3, and 14]. The list of features is shown in Table 1. Pixel-based feature extraction was implemented in MATLAB with 18 features taking the most time.

Table 1: 18 features used in low-level feature extraction

Color-based Features	Mean of Features	Variance of Features
R value	Mean R	Variance R
G value	Mean G	Variance G
B Value	Mean B	Variance B
H Value	Mean H	Variance H
S Value	Mean S	Variance S
V Value	Mean V	Variance V

18 low-level features (K=18) were extracted from each pixel in a given image "I_i" aligned in one long row. Feature Vector called FV was described as follows:

$$FV(I_i) = \{f_1(I) + f_2(I) + \ldots + f_k(I)\}$$

Equation 4

This procedure was applied on all images, including black to not black images, in both training and test images.

$$FVB = \{FV(I_1) + FV(I_{12}) + \ldots + FV(I_n)\}$$

Equation 5

$$FVNB = \{FV(I_1) + FV(I_{12}) + \ldots + FV(I_m)\}$$

Equation 6

In simple terms:

$$FVtrain = \sum_{i=1}^{n} FVB_i + \sum_{j=1}^{m} FVNB_j$$

Equation 7

The same procedure was applied to FV_{test} matrix production. Finally, FVtrain and FVtest were obtained. Pixel-based feature extraction was implemented in MATLAB with 18 features taking the most time. During feature extraction, another important point described in the following paragraphs was considered.

3.3. Feature Vector Normalization

In order to use a classifier, all values in the feature vector must be normalized. This will result in a matrix with values between $[\mathbf{-1}, \mathbf{+1}]$. The normalization step is quite important for removing lighting effects, allowing for better generalization across images [1]. There are several methods used to normalize data in feature vectors in order to remove illumination. The method used in this work subtracts values from the average then divides by twice the standard deviation. The biggest problem with this method is unquestionably SVM normalization. SVMs assume that the data they work with is in a standard range, usually either $[\mathbf{0}, \mathbf{+1}]$or $[\mathbf{-1}, \mathbf{+1}]$ (roughly). Moreover, each dimension of the feature vector should be within this range. Otherwise, if, e.g., dimension 1 is from 0-1000 and dimension 2 is from 0-1.2, dimension 1 becomes much more important than dimension 2, which will skew the results. As a result, the normalization must be applied "by dimension, not instance" prior to sending the $FV_{training}$ to the SVM library. The authors of "*libsvm*" recommend performing a "hard" normalization, mapping the minimum and maximum values of a given dimension to 0 and 1. However, according to Neeraj and his colleagues experiences [2, 3 and 14], a "soft" normalization is better, as mentioned before. These normalized vectors are sent to "*libsvm*" for training. During testing, it is important to construct the test feature vectors in exactly the same way.

$$Normalized\ FV = \frac{FV_i - \overline{FV}}{2 * \sigma(FV)}$$

Equation 8

3.4. Classifier Training

The nature of this project is based on a binary decision. In the near future, the best classifier capable of returning optimal results will be the SVM. As mentioned before, a simple linear approach that is applied to data in a high dimensional space is rendering SVM very useful [8]. Several kernel functions can be used in SVM, including Linear, Quadratic, the Gaussian Radial Basis Function, which is called RBF, Polynomial, etc. Two popular kernel functions were used in this project: "Linear" and "RBF," respectively. The reason these functions were used is attributable to their strong reputation in related papers. It should be noted that SVM requires a lot of data to be well trained, and the more data there is to train, the more trustable the results. As seen in the next chapter, SVM was trained with different amounts of data (200 and 460) and was tested two times using 100 black hair images as well as hair images that were not black.

K – NN was used as the second classifier, and as described before, it works very quickly because of the simplicity of its structure. The whole procedure for SVM was implemented for K – NN as well in order to compare the classifiers.

3.5. Training Precision, Recall, Accuracy

Standard techniques in pattern recognition (and statistics) were employed to gauge how well the designed algorithm works, as well as its rate of accuracy. "For classification tasks, the terms true positives, true negatives, false positives, and false negatives (Type I and type II errors) compare the results of the classifier under test with trusted external judgments. In this study, the mentioned definitions are considered as:

True Positive: Black hair images which are detected as black hair images. *TP*

True Negative: Images that are not of black hair which are detected as such. *TN*

False Positive: Black hair images which are not correctly detected. *FP*

False Negative: Images that are not of black hair which are detected as black hair. *FN*

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN} \qquad Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Equation 9

The chart diagram below describes how several steps of this classification work:

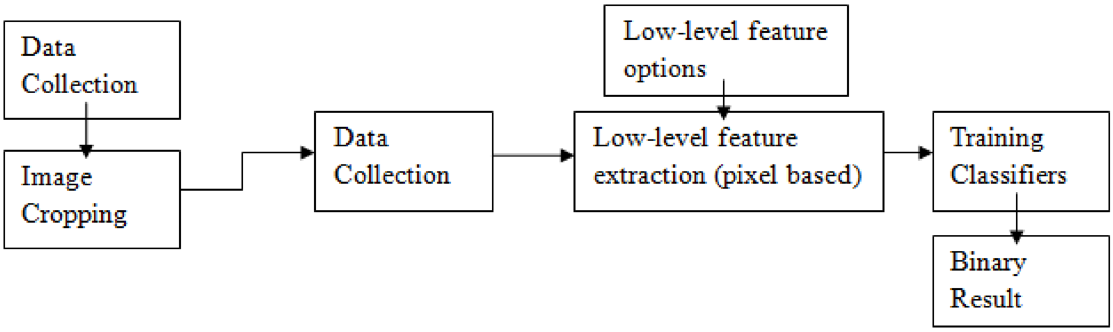

Figure 1: Classification pipeline for hair color detection

4. Results

Different results based on varying parameters were obtained from this experiment, and all results obtained are demonstrated in this report in order to ensure a complete analysis. Some input parameters and classifiers were changed, including number of features, number of data, using the whole and one-fifth of the top of the image, and employing two different classifiers.

4.1. Dataset

Two different datasets comprised of 200 and 460 images, with the first a subset of the second, were used in order to train classifiers. These two datasets allowed for an understanding of the effect of the number of data on a classifier's accuracy.

Figure 2: Samples of dataset used in training as black hair images

4.2. Features

A set of 18 features explained in Table 1 were first used in low-level feature extraction. Secondly, nine features of that set were used in feature extraction.

This change was implemented to understand how much all features have linear independence, which is one of the most important concepts in classifier training. In other words, the features must be independent from other features.

4.3. Images

All features in the previous steps were applied to the whole image in order to define the suitability of these features. To improve the result, a suggestion was proposed. Because hair normally appears on top of an image, feature extraction was done on one-fifth of the top of the image.

Figure 3: Samples of dataset used in training non-black hair images

4.4. Classifier

SVM and K-NN were used as classifiers, and a comparison between results demonstrates their robustness as well as their speed.

In SVM classifier, two different kernel functions, linear and RBF with sigma = 1, were employed; however, all results generated by the RBF function were invalid. In total, training of classifiers was repeated 16 times, and different sets of results were obtained, including precision, recall, accuracy and processing time.

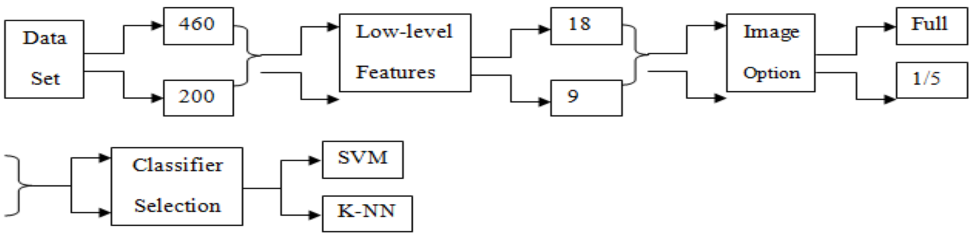

Figure 4: The pipeline above shows how data were prepared to train SVM and K-NN classifiers.

4.5. Data Analysis

All results categorized by the mentioned parameters are displayed in Table 2. The results are divided into two sets: those with an acceptable accuracy rate and those with a lower than acceptable accuracy rate.

Table 2: The results are shown below for different features and classifiers (FV stands for feature vector image, and full means all samples were used).

No. 1				No. 2			
FV = 18				FV = 18			
Image Full				Image Full			
Data460				Data200			
Function	SVM linear	SVM RBF	KNN	Function	SVM linear	SVM RBF	KNN
Precision	0.95	NaN	0.74	Precision	0.74	NaN	0.68
Recall	0.959596	NaN	0.5873	Recall	0.601626	NaN	0.56667
Accuracy	0.955	NaN	0.61	Accuracy	0.625	NaN	0.58
Time	15.302646	36.90137	40.7	Time	9.5130642	12.87481	30.4234

No. 3				No. 4			
FV = 18				FV = 18			
Image 1/5				Image 1/5			
Data460				Data200			
Function	SVM linear	SVM RBF	KNN	Function	SVM linear	SVM RBF	KNN
Precision	0.96	NaN	0.74	Precision	0.74	NaN	0.68
Recall	0.96	NaN	0.5873	Recall	0.601626	NaN	0.56667
Accuracy	0.96	NaN	0.61	Accuracy	0.625	NaN	0.58
Time	6.9152566	7.213724	13.7323	Time	3.0438873	2.779685	5.68711

No. 5				No. 6			
FV = 9				FV = 9			
Image Full				Image Full			
Data460				Data200			
Function	SVM linear	SVM RBF	KNN	Function	SVM linear	SVM RBF	KNN
Precision	0.99	NaN	0.75	Precision	0.6	NaN	0.66
Recall	0.9519231	NaN	0.5814	Recall	0.5454546	NaN	0.5641
Accuracy	0.97	NaN	0.605	Accuracy	0.55	NaN	0.575
Time	13.466941	15.21303	31.7073	Time	4.7497586	5.690107	13.7967

No. 7				No. 8			
FV = 9				FV = 9			
Image 1/5				Image 1/5			
Data460				Data200			
Function	SVM linear	SVM RBF	KNN	Function	SVM linear	SVM RBF	KNN
Precision	0.99	NaN	0.75	Precision	0.6	NaN	0.66
Recall	0.9519231	NaN	0.5814	Recall	0.5454546	NaN	0.5641
Accuracy	0.97	NaN	0.605	Accuracy	0.55	NaN	0.575
Time	3.5236512	3.333086	6.50528	Time	1.8429846	1.762057	3.38216

The first logical idea regarding the number of data, as mentioned previously, is that the more data there are to train the classifier, the more accurate and reliable the results. In this work, the same result was achieved: an increase in accuracy followed an increase in the number of data. When 460 data were used, highly acceptable results in terms of SVM accuracy were obtained. Conversely, when less data (200 images) were used to train SVM, less accurate results were achieved.

Since strong results were achieved with 460 data in both classifiers (SVM and K-NN), this may well verify the first dataset used for training. Because a lower accuracy rate was achieved when 200 data were employed in both classifiers, it could justify the concept above. As is well-known in pattern classification, the training data must be shuffled to then be sent for training. This procedure must be repeated several times, and the best trained network (SVM in this instance) could be used as the best one (all statements are valid for test data as well). Furthermore, the cross validation process used to identify the most trusted classifier must be experienced.

Unfortunately in this work, all data were shuffled once, and no cross validation was done. Feature extraction and the training of SVM and K-NN classifiers took a lot of time. Thus, data shuffling and cross validation were not done because of time constraints. In future work, it may be possible to carry out all mentioned steps in order to obtain a more trusted result.

Another solution to make classifiers more reliable is to train them with thousands of data derived from different databases. The more data viewed by a classifier, the more reliable the classifier will be. However, it should be noted that most classifiers could not extrapolate data.

Table 3: Accuracy of classifiers in terms of dataset and classifiers

SVM 460	SVM 200	K-NN 460	K-NN 200
0.955	0.625	0.61	0.58
0.96	0.625	0.61	0.58
0.97	0.55	0.605	0.575
0.97	0.55	0.605	0.575

To sum up, both classifiers (SVM and K-NN) showed better results with a larger training dataset over a smaller training dataset. As was expected, the results derived from SVM were more accurate than those derived from K-NN. The justification for this stems from the complexity in SVM and K-NN structures, as discussed in related information. Much experience has proven that SVM is a more powerful classifier in pattern classification, and better results with this method are invariably expected. As mentioned before, K-NN is a faster approach (simpler is likely more accurate) for gaining an idea about data and their groups.

$$data \uparrow \sim accuracy \uparrow \sim system\, reliability \uparrow$$

$$In\ terms\ of\ accuracy\ and\ performance: SVM > K-NN$$

Equation 10

The second parameter examined was low-level features. The most important step in this work was to find suitable features. As mentioned before, according to the extant literature, 18 features were identified based on color values in two different color spaces. On the other hand, as is well-known, the selected features must be linearly independent. The problem with feature independence stems from the existence of well-separated features used for classifier training. For the first set of features, RGB values with their averages and variances were selected. In order to decrease the volume of feature space, and in the hope of identifying more independent features, the average and the variance of RGB were eliminated from the feature vectors in the second set of features, and nine features were selected based on RGB values, mean and variance of HSV.

In summary, different results were achieved. When SVM were trained by the larger dataset, they returned a slightly better result with less features (nine features). However, the SVM trained by the smaller dataset demonstrated a slightly worse result with less features. It is obvious that SVM in conjunction with fewer features and less training data returns worse result because it has neither sufficient training data nor sufficient features assisting in its training. It should be noted that the feature vector changes from 460*201*151*18 = 251306280 to 200*201*151*9 = 125653140. In summary, low data numbers do not provide enough information to properly train linear and RBF SVM.

A small increase was observed in SVM trained by a larger dataset and K-NN trained by both datasets. Regarding K-NN, it can be said that since it has a significant sensitivity to linearly independent inputs, and the more independent they are, the better trained K-NN will be. The improvement in the K-NN result in this portion is derived from a well-separated feature vector compared to the larger feature space.

The same idea could justify the result of SVM, as it was trained by more accurately selected features, and the number of the training data remained at a healthy amount (251306280). However it may also indicate that because of the internal structure of SVM, it does not have the same amount of sensitivity to linearly independent inputs compared to K-NN, because SVM has the ability to ignore useless features, or it can decrease the weight of less useful inputs (which is not precisely found in K-NN).

Table 4: The comparison between the number of feature vectors, datasets and classifiers

SVM	FV 18	FV 9	KNN	FV 18	FV 9
460	0.955	0.97	460	0.61	0.605
460	0.96	0.97	460	0.61	0.605
200	0.625	0.55	200	0.58	0.575
200	0.625	0.55	200	0.58	0.575

As mentioned in the methodology chapter, all selected features were applied to the whole of the image, and a pixel-based feature extraction was applied to all images. On the other hand, there are several ways to improve the obtained results, such as independent component analysis (ICA) or increases in the dataset number. Meanwhile, an idea derived from hair location on the face was realized in this work as a developing method to obtain a better result.

As is well-known, hair appears on the top of the head, and it occupies almost one-fourth or one-fifth of the top of the head. Thus, the second time, feature extraction was done on one-fifth of the top of all images in the datasets, and the results were analyzed.

A slightly better result was recognized during this process but there were insignificant changes in accuracy. This demonstrates that the features used in this classification work very well and they do not belong on one part of the image. The other more important concept is that these well-trained classifiers are working on the most important parts of the image, and they do not use all pixels to find a black hair image or a non-black hair image. If significantly better result had been achieved, it could be said that the features would not have been well selected. A small improvement in accuracy was expected and was obtained.

As indicated in the previous chapters and charts, two classifiers were trained in order to provide ideas on which one could be more precise and what advantages and disadvantages exist. In fact, three classifiers were used: Linear SVM, Radial Basis Function SVM (RBF), and K-NN.

In the RBF experience, the value of sigma was one ($\sigma = 1$), and no acceptable result was obtained. Since this work is a short-term project, the only one value for K-NN was given. In practical terms, K-NN must be trained by several Ks (cross validation), and the best result would be considered the answer. Here, K = 1 was taken.

The figure below shows the "Recall – Precision" curve for SVM and KNN, indicating that all SVM results were significantly better than K-NN, and greater variety could be seen in the SVM results. As expected, SVM showed more power and accuracy in the peer conditions.

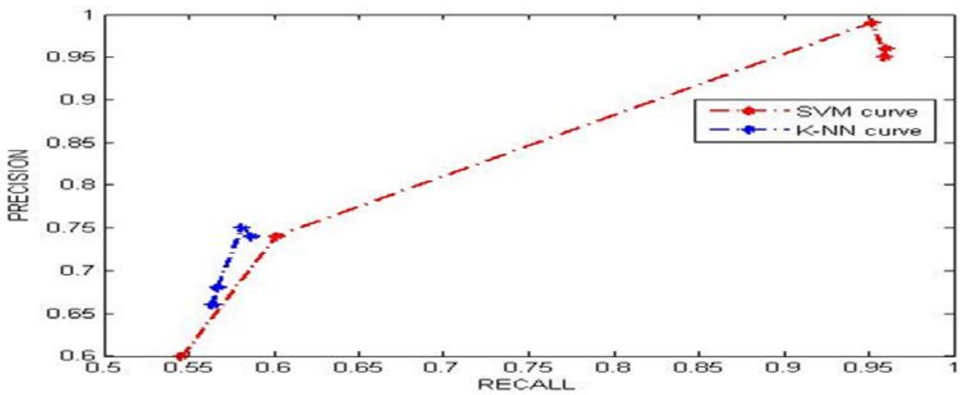

Figure 5: SVM and K-NN Recall – Precision curve

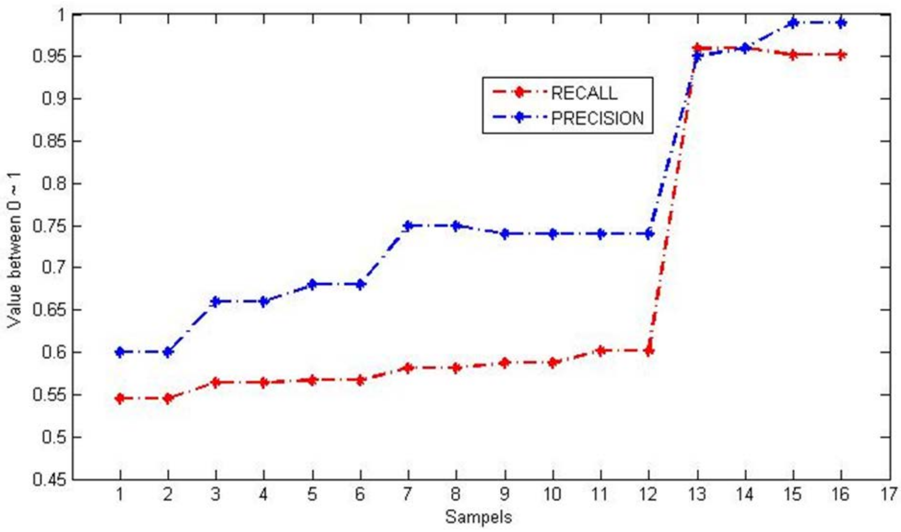

Figure 6: Recall and Precision Values (Vertical axis), Samples (Horizontal axis)

Table 5: Experience and associated information (Feature Vectors dimension, Data size, Classifiers)

No.	FV	Image	Data	SVM / K-NN	No.	FV	Image	Data	SVM / K-NN
1	9	Full	200	SVM	9	18	Full	460	K-NN
2	9	one-fifth	200	SVM	10	18	one-fifth	460	K-NN
3	9	Full	200	K-NN	11	18	Full	200	SVM
4	9	one-fifth	200	K-NN	12	18	one-fifth	200	SVM
5	18	Full	200	K-NN	13	18	Full	460	SVM
6	18	one-fifth	200	K-NN	14	18	one-fifth	460	SVM
7	9	Full	460	K-NN	15	9	Full	460	SVM
8	9	one-fifth	460	K-NN	16	9	one-fifth	460	SVM

Additionally, the accuracy values delineated below demonstrate which experiment was the most successful. This graph show that the best result was achieved in samples #15 and 16, which represent the SVM trained by 460 data and nine features on the whole of images and the SVM trained by 460 data and nine features on one-fifth of the top of images, respectively. These results were expected. The most accurate experiment had a 99% accuracy rate, and the worst result remained at a 55% accuracy rate (both will be discussed in the "Comparison to Reference Papers").

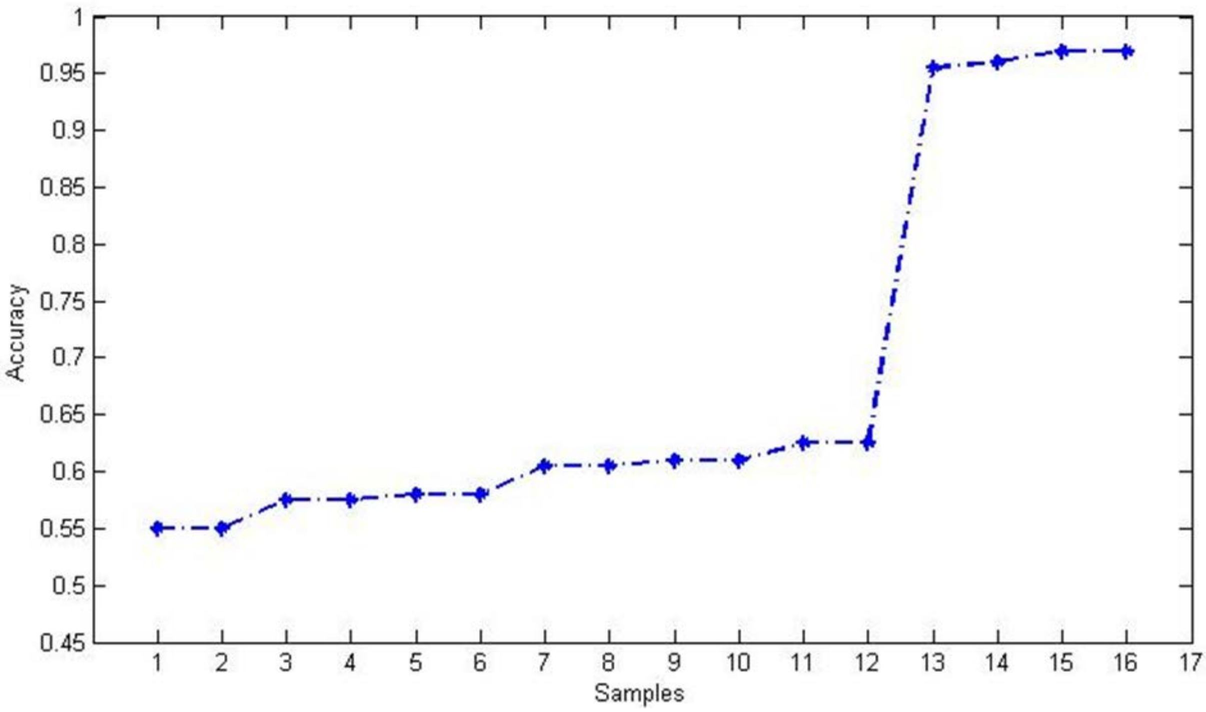

Figure 7: Accuracy of classifiers in terms of 16 different experiences

The lowest accuracy rate occurred in the SVM trained by 200 data and nine features on the whole of images and one-fifth of the top of images, respectively.

5. Related works

Two critical parameters which must be compared to reference papers are the number of data used in this project and the accuracy of the trained classifiers. The exact number has not been mentioned in two reference papers [1,2] but they have used thousands (two or three thousand) of images in order to train and test their classifiers. Therefore, with the use of this huge number of data, the results could be quite reliable. Two accuracy rates have been reported in the reference papers: 94.56% and 90.8%. The datasets used in this report are about 660 and the best result recorded in this report had a 97% accuracy rate, compared to a 55% accuracy rate in the worst situation. One good question is why the better result and, conversely, why the worst result? The principal answer to these questions can be derived from the datasets. The datasets used here were selected randomly on the Internet and different databases, and they had a lot of restrictions in terms of diversity and variety. As a result, it is quite possible that in the best result, there would be a high level of similarity between the training data and the test data. In the worst situation, the classifier did not encounter enough diverse data, especially for non-black hair images. One weakness here could be the versatility of data. Because of time limits and the time-consuming nature of data preparation, the dataset used was not quite ideal. If the average is taken for precision, recall and accuracy of these results, the following holds.

The above table justifies that the reliable precision of this work is $82.125 \pm 3\,\%$ which almost 85%, and this could represent an acceptable rate of precision compared to almost 91% in the reference paper. On the other hand, this table shows that the recall rate is not as accurate as the rate of precision. This shows that the non-black hair data could not train the classifier very well, and this is obvious, as mentioned previously. Because of the vast diversity of non-black hair colors, the datasets used did not include all possible colors for non-black hair.

Table 6: Mean and variance of performance parameters

SVM	MEAN	VAR
Accuracy	0.775625	0.111398
Precision	0.82125	0.029127
Recall	0.77144782	0.042222

6. Image Analysis

In the following figures, some sets of images are or are not truly distinguished by SVM. The similarities between black hair samples that are well-detected is based on color, and the similarities between black hair not distinguished as black hair is a semi-rotated face ignoring one eye and face normalization. The non-black hair that is well-detected had enough resembling data in the training set, and the non-black hair images that were inaccurately detected were poor in terms of data similarities in the training dataset. This procedure was done manually by searching the dataset and examining all images.

Figure 8: Black hair images correctly detected

Figure 9: Black hair images not correctly detected

Figure 10: Non-black hair images correctly detected

Figure 11: Non-black hair images incorrectly detected

7. Conclusion

Low-level feature extraction is one of the best approaches that can help us to apply most pattern recognition methods. Today, the super vector machine is employed as a robust method in almost every classification project. This classifier provides the highest performance with different kernel functions, covering all needs in object detection and other subjects.

In face detection, many features have important roles, and one of them is hair (recognition). From the past to the present, this feature has not been studied very much, which is due in part to its complexity. This report will implement a practical algorithm in order to improve upon black hair classification. Hair color (especially black hair) helps us in gender classification and face identification. Although black hair detection is not a simple job, acceptable results can be obtained by spending more time on it. The results in this course project were approximately close to results obtained in reference papers, which reveal to us that the right direction has been chosen. For future works, it is highly recommended to apply some pre-processing tasks in order to automatically identify the hair region and then extract features from the detected area on the head as hair. This process may provide a more accurate feature vector. In simple terms, a combination of a geometrical approach and low-level (high-level) feature extraction could yield much stronger results.

If the hair region can be detected, the use of frontal only facial images is not needed, and some profile images could be used in training and testing of the dataset. This result might offer the possibility of using black (or other colors) hair for real-time facial identification and recognition.

For future work, in the feature extraction step, it could be possible to calculate other features in three channel colors, which might provide a better feature space for training the classifier. Although some related works have been published, more projects on (black) hair classification and detection will be expected in the near future. Additionally, the new cutting-edge Deep Learning technology could open new avenues in face recognition and classification.

References

[1] Kumar, Neeraj, Peter Belhumeur, Shree Nayar, "Facetracer: A search engine for large collections of images with faces," in In European conference on computer vision, pp. 340-353. Springer Berlin Heidelberg, 2008.

[2] Neeraj Kumar, , Alexander Berg, Peter N. Belhumeur, and Shree Nayar, "Describable visual attributes for face verification and image search," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. no. 10, pp. 1962-1977, 2011.

[3] Yacoob, Yaser, and Larry Davis, "Detection, analysis and matching of hair," in In Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, vol. 1, pp. 741-748. IEEE, , 2005.

[4] Diedrick Marius, Sumita Pennathur, and Klint Rose, "Face detection using color thresholding and eigenimage template matching," Digital Image Processing project, Stanford University, 2003.

[5] Ion Marques, Supervisor: Manuel Grana., "Algorithms, Face Recognition. "Proyecto Fin de Carrera"," June 16, 2010.

[6] Zheng Ji, Bao-Liang Lu, and Xiao-Chen Lian, "Gender classification by information fusion of hair and face," INTECH Open Access Publisher, 2009.

[7] Steve Lawrence, C. Lee Giles, Ah Chung Tsoi, and Andrew D. Back, "Face recognition: A convolutional neural-network approach," IEEE transactions on neural networks , vol. 8, no. no. 1, pp. 98-113, 1997.

[8] Kurt Hornik, David Meyer, and Alexandros Karatzoglou, "Support vector machines in R," Journal of statistical software , vol. 15, no. no. 9, pp. 1-28, 2006.

[9] Jason Weston, "Support vector machine (and statistical learning theory) tutorial," NEC Labs America, 1998.

[10] Elham Bagherian, Rahmita Wirza OK Rahmat, "Facial feature extraction for face recognition: a review," in In 2008 International Symposium on Information Technology, IEEE, Malaysia, 2008.

[11] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, Shree K. Nayar, "Attribute and simile classifiers for face verification," in In 2009 IEEE 12th International Conference on Computer Vision, pp. 365-372. IEEE, 2009., 2009.

[12] Saman Sarraf, Cristina Saverino, Halleh Ghaderi, John Anderson, "Brain network extraction from probabilistic ICA using functional Magnetic Resonance Images and advanced template matching techniques," in Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference, Toronto, 2014.

[13] Saman Sarraf , Cristina Saverino, Ali Mohammad Golestani, "A Robust and Adaptive Decision-Making Algorithm for Detecting Brain Networks Using Functional MRI within the Spatial and Frequency Domain," in The IEEE International Conference on Biomedical and Health Informatics (BHI) , Las Vegas, 2016.

[14] Saman Sarraf, Jian Sun, "ADVANCES IN FUNCTIONAL BRAIN IMAGING: A COMPREHENSIVE SURVEY FOR ENGINEERS AND PHYSICAL SCIENTISTS.," International Journal of Advanced Research, vol. 4, no. 8, pp. 640-660, 2016.

[15] Saman Sarraf , Ehsan Marzbanrad , Hamid Mobedi, "Mathematical Modeling for Predicting Betamethasone Profile and Burst Release From In Situ Forming Systems Based On PLGA," in Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference on, Toronto, 2014.