NAMING FACES ON THE WEB

A THESIS
SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Hilal Zitouni
July, 2010

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Y. Doç. Dr. Pınar Duygulu Şahin(Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Fazlı Can

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Aydın Alatan

Approved for the Institute of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Institute

ABSTRACT

NAMING FACES ON THE WEB

Hilal Zitouni
M.S. in Computer Engineering
Supervisor: Asst. Prof. Dr. Pınar Duygulu Şahin
July, 2010

In this study, we introduce a method to name less-frequently appearing people on the web via naming frequently appearing ones first. Current image search engines are widely used for querying a person, however; retrievals are based on textual content; therefore, the results are not satisfactory. On the other hand, although; face recognition is a long standing problem; it is tested for limited sizes and successful results are acquired just for face images captured under controlled environments. Faces on the web, contrarily are huge in amount and vary in pose, illumination, occlusion and facial attributes. Recent researches on the area, suggest not to use simply the visual or textual content alone, but to combine them both. With this approach, face recognition problem is simplified to a face-name association problem.

Following these approaches, in our method textual and visual information is combined to name faces. We divide the problem into two sub problems, first the more frequently appearing faces, then the less-frequently appearing faces on the web images are named. A supervised algorithm is used for naming a specified number of categories belonging to more frequently appearing faces. The faces that are not matched with any category are then considered to be the less-frequently appearing faces and labeled using the textual content. We extracted all the names from textual contents, and then eliminate the ones used to label frequently-appearing faces before. The remaining names are the candidate categories for less-frequently appearing faces. Each detected less-frequently appearing face finally matched to the names extracted from their corresponding textual content. In order to prune the irrelevant face images, finally, the most similar faces among this collection are found to be matched with their corresponding category.

In our experiments, the method is applied on two different datasets. Both

datasets are constructed from the images captured in realistic environments, varying in pose, illumination, facial expressions, occlusions and etc. The results of the experiments proved that the combination of textual and visual contents on realistic face images outperforms the methods that use either one of them. Besides, handling the face recognition problem as a face-name association, improves the results for the face images collected from uncontrolled environments.

Keywords: face recognition, face detection, face retrieval, face naming, naming faces, SVM, SIFT.

ÖZET

WEB ÜZERİNDE GÖRÜLEN YÜZLERİN İSİMLENDİRİLMESİ

Hilal Zitouni
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Y. Doç. Dr. Pınar Duygulu Şahin
Temmuz, 2010

Bu çalışmada, web üzerinde sık görülen yüzlerin isimlendirilmesinden faydalanarak, nadir görülen yüzlerin isimlendirilmesini sağlayan bir yöntem sunulmuştur. Mevcut görüntü arama motorları, günümüzde yaygın olarak bir kişinin sorgulanması için kullanılmaktadır; ancak sonuçlar, görüntülere ait metinler kullanılarak getirilmektedir, dolayısıyla yüksek başarı elde etmek için bu yöntem yetersiz kalmaktadır. Yüz tanıma sistemleri ise, üzerinde uzun zamandır çalışılan bir konu olmasına rağmen, başarılı sonuçlar sadece kontrollü ortamlarda çekilen fotoğraflar üzerinde, limitli bir veri kümesi için elde edilmiştir. Web üzerinde bulunan yüz resimleri ise, bunun aksine, gerçek ortamlarda çekilmiş olup, pozisyon, ışıklandırma, yüz ifadeleri vb. farklılıklar göstermektedir. Bu alanda yapılan son çalışmalar ise, sadece metin ya da sadece görsel içeriği kullanmaktansa, ikisini birleştirerek daha başarılı sonuçlar elde etmeyi önermektedir. Bu yaklaşım kullanılarak, yüz tanıma problemi, isim-yüz eşleştirmesi olarak basitleştirilmiştir.

Yukarıda bahsetmiş olduğumuz bilgiler doğrultusunda, yüzleri isimlendirmek için geliştirmiş olduğumuz metot, metinsel ve görsel içeriği birlikte kullanmıştır. Problem, iki alt probleme bölünmüş olup, ilk basamakta sık görülen yüzler isimlendirilmiş, ikinci basamakta ise nadir görülen yüzler isimlendirilmiştir. Sık görülen yüzlerin isimlendirilmesinde, belirli bir kategori sayısı için güdümlü algoritma tekniği kullanılmıştır. Bu aşamada isimlendirilemeyen yüzler, nadir görülen yüz olarak nitelendirilmiş ve bu yüzlerin isimlendirilmesinde ise resme ait metinlerden faydalanılmıştır. Metinsel içeriklerdeki tüm isimler çıkarılmış, bunların arasından ilk aşamada sık görülen bir yüz ile eşleştirilmiş olan isimler elenmiştir. Geriye kalan isimler, resimlerdeki nadir görülen yüzler için aday isim

kategorilerini oluşturacaktır. Her nadir görülen yüz, ilgili metinden çıkarılan isimlerle eşleştirilmiştir. Son olarak, bir isim için toplanmış yüzler arasındaki alakasız resimlerin elenmesi adına birbirine en çok benzeyen resim topluluğu bulunarak, bu topluluktaki yüz resimleri, altında toplanmış oldukları isim ile eşleştirilir.

Deneylerimiz boyunca, önerilen metot iki ayrı veri kümesi üzerinde uygulanmıştır. İki veri kümesi de, kontrolsüz, gerçek ortamlarda çekilmiş olan fotoğraflardan oluşup, pozisyon, ışıklandırma, yüz mimikleri açısından farklılıklar göstermektedir. Deneylerimizin sonuçları kanıtlamıştır ki, metin ve görsel içeriğin birarada kullanılması, ikisinden sadece birinin kullanıldığı metotlarlardan daha başarılı sonuçlar elde edilmesini sağlamıştır. Bunun yanı sıra, yüz tanıma problemini, isim-yüz eşleştirme problemi olarak basitleştirmek, kontrolsüz ortamlarda çekilmiş olan fotoğraflar üzerinde de başarı elde edilmesine yol açmıştır.

Anahtar sözcükler: yüz tanıma, yüz bulma, yüz sorgulama, yüz isimlendirme, SVM, SIFT.

To my father...

Acknowledgement

I would like to thank my supervisor Prof. Dr. Pınar Duygulu for her support, guidance and encouragement. It was an honour to work with her.

This research is partially supported by TÜBİTAK Career Project, grant number 104E065.

I want to express my thanks to my family, especially to my mother, for their endless love and support during this period.

I would like to thank my boss Nilüfer İnce, for her valuable support, help and understanding especially during the last period of my master's thesis.

I would like to thank all my friends for being supportive and helpful during my thesis study. I would like to thank my colleague Seza Soyluçiçek for her graphical contribution on the design of my diagrams and figures.

I would like to express special thanks to my friend Melih Özbekoğlu, who have been very supportive and helpful, technically and psychologically during the entire period of my thesis study. And, I want to express how greateful I am to have a friend like him, it would be harder to get through this period without his support.

Last but not least, I would like to express my deepest appreciation and thanks to my present and future life partner Barış, for being so nice, understanding and supportive during my entire master's thesis. Without his support and understanding it would be really hard to finish this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The amount of data on the web has been increasing tremendously, resulting in a demand for efficient and effective retrieval systems.

Searching for people is a desired and common task especially for news related pages. The usual approach is to query for the name of a person in the surrounding textual content e.g in the captions of news photographs. However; such an approach is likely to return irrelevant results (see Figure 1.1). Since there may be several people or no people at all in the image associated with the textual content including the name.

In order to get the most desired results, the visual information should also be taken into account. Although face recognition seems like a solution to the problem, it is not easy to detect and recognize faces on photographs which are taken in natural environments.

The traditional face recognition systems are successful only when faces are frontal and captured in controlled environments. However, faces on the web differ in illumination, pose, size, and there are several other factors making it difficult to recognize people even by humans, such as occlusion, aging, clothing and make-up (see Figure 1.2). Therefore, most of the current face recognition systems are barely successful for faces captured in realistic environments.

Figure 1.1: Sample images collected from the first pages of Google image search results for the text based query on *George Bush*.

Recently, rather than using purely text based, or purely visual based methods, textual and visual information are integrated [20], [23], [21]. Having the visual information and the possible names appearing around the corresponding image, the problem of face recognition is simplified to finding face-name associations.

Inspired by those studies, in this study, we propose a method for naming faces in the news photos appearing on the web, as in Yahoo News, etc. With the light of the information that some people mentioned in the news photos appear more frequently than the others, naming frequently appearing faces will be a relatively easier task, compared to naming less frequently appearing faces on the photos.

In this study, we aim to address this challenge, and consider also naming people appearing less frequently, who appear on the web next to people appearing more frequently. The steps of our algorithm will be as follows:

- Name more frequently appearing people on the Web
  - Label the faces with the name list of more-frequently appearing people via a supervised classification
  - Find the outliers (in other words the faces that are not in the list of more frequently appearing people)
- Name less frequently appearing people on the web
  - Assign names to outliers detected in the previous step using textual content
  - Prune the irrelevant face images from the images matched with the name categories generated for the outliers.

The second step of our methodology, naming less frequently appearing people is illustrated in Figure 1.3.

The thesis will be organized as follows; first the related work on the relevant subjects will be discussed in Chapter 2. The details of the steps of this algorithm will be explained in Chapter 3. Chapter 4 will give brief information on the datasets used, and the corresponding experimental results will be presented in Chapter 5. The evaluations of the results will be discussed in Chapter 6. Finally, the conclusion and future work will take place in Chapter 7.

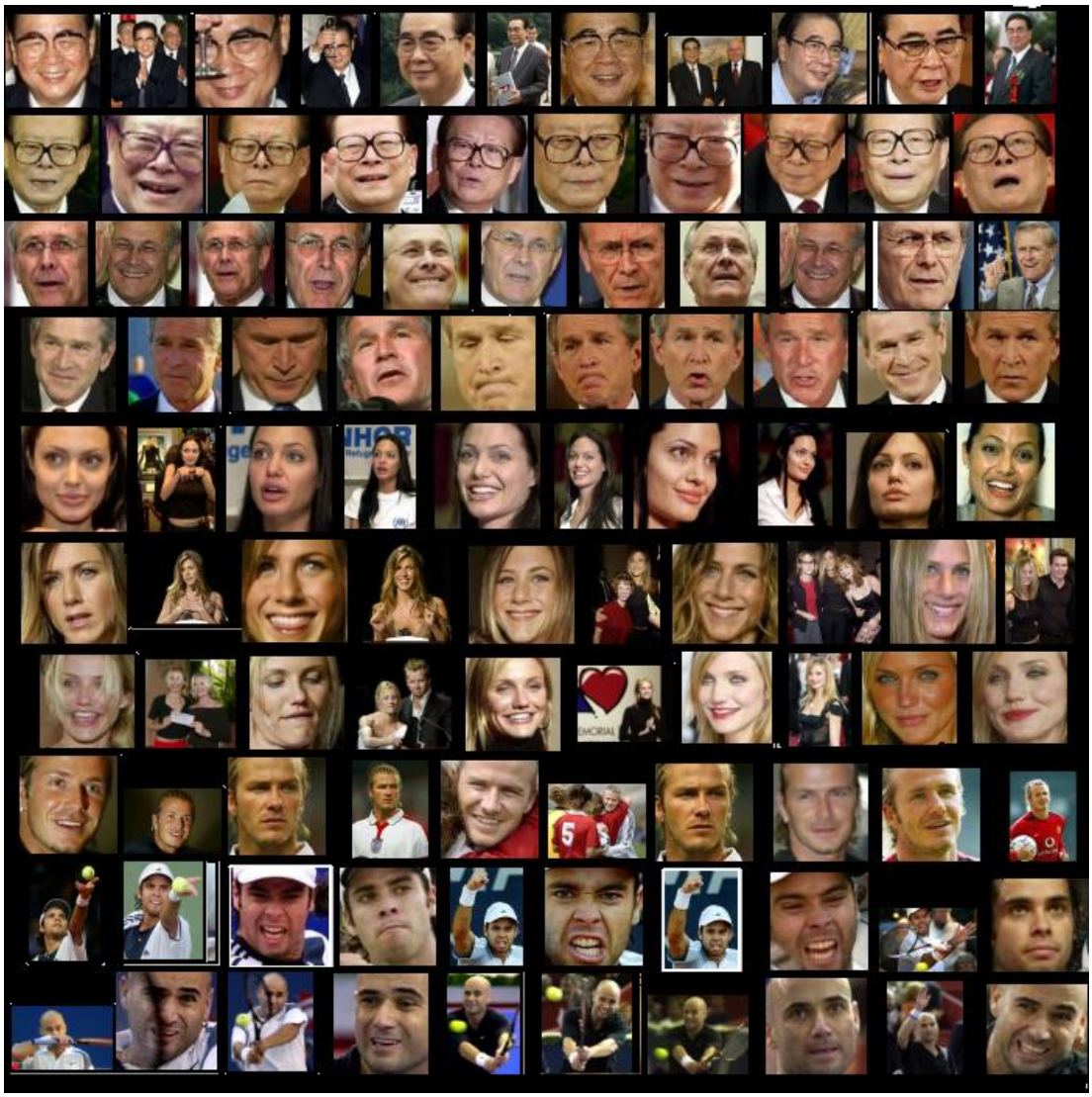

Figure 1.2: Faces collected under uncontrolled environment for Donald Rumsfeld. Images are from Faces in the Wild Dataset [3]. Each row corresponds to a different person. Note the large variety in pose, illumination, expression and make-up.

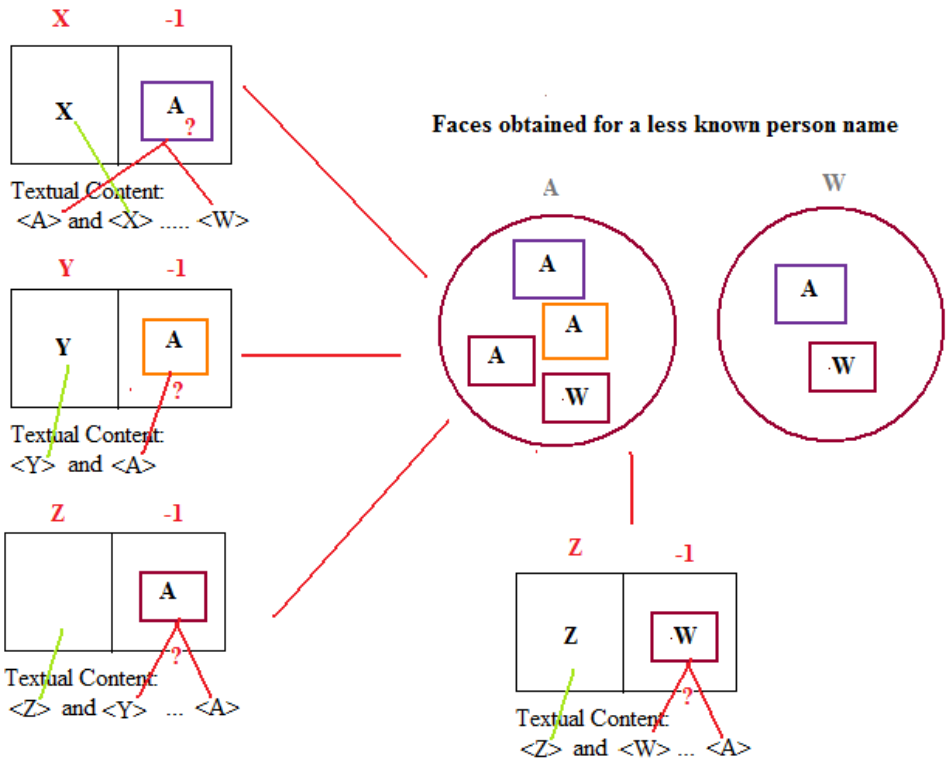

Figure 1.3: The outlier faces (images with label *-1*) detected while naming faces, will be matched with the names extracted from the textual content. Each name extracted from the textual contents will then be matched with less-known (less frequently appearing) faces, and as a result, for an extracted name bunch of face images will be obtained.

Chapter 2

Related Work

Face Recognition is a well-studied and challenging problem that receives significant attention for the scientists on the area of image processing, computer vision, pattern recognition, machine learning, neural networks and etc. So far, it begins to be a considerable need for technological achievements especially in commercial and law enforcement applications. People tend to improve methods for applications that need security and privacy rather than using passwords or pins. Although different types of personal biometric identification exists, such as retinal scan or fingerprints, as stated in the study of Zhao et al. [28] , those methods rely on the participation of the subject of person. Therefore, with the expanding demand on technological improvement, face recognition is about to become an integral part of our lives.

Although it is a long-studied problem, the solution to the problem is not yet achieved adequate results. Recognizing a detected face on an image is difficult due to the variations in positions, illumination, orientations, occlusions, facial expressions and etc. Therefore, existing face-recognition systems use faces captured in controlled environments.

Zhao et al. [28] in their survey of face recognition, divide the problem into three steps as illustrated in Figure 2.1. Given an input image or video scene, the faces are detected as a first step, then the features on the detected regions

are extracted, finally the face is recognized using the extracted facial features. In this survey, in order to emphasize the importance of the face recognition problem, and for an entire comprehension, the studies over 30 years have been examined throughout the aspects of psychophysicists and neuroscientists. Some of the issues considered relevant to the face recognition problem concerned by psychophysicists and neuroscientists are as follows:

- Is face recognition a dedicated process?
- Is it a holistic or feature analysis?
- Which facial features are more significant in recognition?
- How significant are the caricaturist and distinctive facial features in recognizing faces?

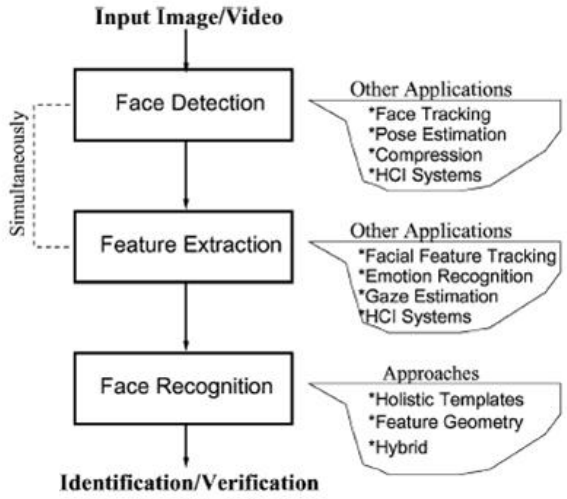

Figure 2.1: Three steps of the Face Recognition Problem (taken from [28]).

Examining through the face recognition problems, the solutions proposed require a dataset of images captured under controlled environments. Meanwhile, as the need for the face recognition grows, existing face recognition systems are

no more adequate. There are a huge amount of face images on the web, and relevant retrieval of a specific face image search among this huge pool is significantly demanded. However, the face images on the web are not appropriate to be the inputs for a face recognition system, since they are captured in a realistic environment, which may result in variations in pose, illumination, occlusion and etc. In order to recognize faces from such images, a different perspective is proposed by the researchers. Rather handling the problem as a face recognition problem, they come up with the idea of name-face association using the textual contents of the face-images. Regarding this approach, name-face association is first studied by Satoh et al. in 1997 [23]. Textual and visual information from videos are collected in order to extract a face-name association. The association is matched by extracting the names from scripts and faces from frames that appear at the overlapping time periods. Following this extraction the co-occurrence factor C (N, F) is calculated by finding the occurrence rate of a face F around a name in videos, in order to determine the best N associated faces for a name. The most similar face in the association set with the dataset of that name is decided as the associated resulting face for that name.

Although naming faces is not considered as a face recognition problem in general, Liu et al. in their study [18] claims that both of them are the same problem. First, the face data is collected from web search engines, and the correct images among those images are considered to be the recognition dataset. After forming the dataset, the naming faces problem is a face recognition problem, hence the faces are detected on images, and the representation for the detected faces Gabor feature approach [25] is adopted, finally with a threshold value approach the face matching procedure is completed.

Recent researches show that combining both textual and visual information increases the accuracy of face-name association [27], [13]. Based on this assumption, Berg et al., propose a method for naming faces in their large dataset of images that are taken in uncontrolled environments. Using simple natural language and clustering techniques on the images from news with their associated captions, the faces are named [13]. The entire dataset of face images are put into a pool to be clustered according to their names. However, in case of a small

variation in environment conditions, this approach may produce poor results.

Ozkan et al. in [21] apply a graph-based method for naming faces. To construct their similarity graph they combine both textual and visual information, as well. In their graph, the nodes represent the images and the edges represent the weight of similarity between the nodes. The similarity is calculated by comparing the interest points between two faces. Unlike existing face representations with interest points, not only particular points on the face are used, but also other detected interest points are taken into account. On the other hand, the use of all detected interest points on a face, rather then particular interest points, do not always give the best result. Finally, using their similarity graph, the most similar images are found via producing the densest subset on that graph, which leads the irrelevant images to be eliminated.

Guillaumin et al. in their study [10] propose a method similar to [6]. They name faces in their database that consists of news photos with captions by considering two scenarios, one is to find faces for a single query, and the other is to name all the faces in their database. In order to achieve the first scenario, they apply the method explored in Ozkan. et al. [21] and find the densest set for a single query. For the second scenario, which is to name all the faces in their database, two approaches are used on a graph based method. They too, construct a similarity graph where the nodes represent the images and the edges represent the similarity weight, however, their similarity weights were different from Ozkan et al. They preferred to select 9 facial features rather than using all the extracted interest points. The first approach is a kNN method with a threshold and the second approach is to differentiate between the neighbors. With the use of 9 particular facial features, they overcome the matching interest points problem encountered in [6].

Satoh et al in [16] introduce an unsupervised method for annotations of faces on the web. Their method consists of two steps, where the first step is to mine the data from the web and find the densest set via ranking the distribution of visual similarities and the second step is the classification of the output query where binary labeling of faces as desired person or non-desired person is determined.

The ranking list of the faces is estimated by their Rank-By-Local-Density-Score method, which is calculated with LDS (p, k); the average distance of a point p to its k-nearest neighbors. The labeling process in the second step is strengthened via Ranking by Bagging of SVM Classifiers method. This method is improved using SVM with probabilistic output, so-called LibSVM [5]. Using a density based estimation, they find their density set, and unlike Ozkan et al. this method does not require a threshold value.

Berg et al, in their study Names an Faces [3], propose a method for association of names to faces using a more realistic dataset which is different from existing face recognition dataset in the sense of faces being captured "in the wild". Dataset for general face recognition should be captured in a controlled environment, for the recognition to be accurate. However, Berg et al, automatically constructed their face data set from uncontrolled environments which have a wide range of positions, illuminations and poses. Their dataset consists of 30.281 face images which are detected from half a million captioned new images.

One recent study on name-face association is proposed by Phi The Pham et al. In their study [22] also focused on name-face association; however rather than the general approach of assigning names to the faces on the images, they propose a method which aims to achieve a one-to-one assignment for names and faces. In order to do so, they proposed three significant models, one is to assume the names in the texts generate the images, the second one is to assume the faces on the images generate the names, and the last one computes the alignment of names and faces with a joint probability calculations of names and faces, P(f,n). For each image-text pair si, with F_i faces and N_i names, there are several alignment schemes aj. In order to achieve a successful one-to-one alignment they used a standard Expectation Maximization algorithm, where their expectation step is to estimate the likelihood of eacg alignment aj, for si; and the Maximization Step updates the probability distributions using the alignment estimations. IN order to strengthen their strategy to find alignments, they use two additional scores, picturedness and namedness, to be used in P(f—n) and P(n—f) calculations. They come up with this idea based on the case that not all the names extracted from texts, or not all the faces extracted from images, have the same possibility for

being significant in terms of having a corresponding face-image just because the name is mentioned, picturedness, or having a corresponding name just because the image contains the face, namedness. The picturedness and namedness scores are then used for the evidence of name-face co-occurance. Finally, for each story si, the alignment aj with the highest corresponding aigma ij obtained from the E-step of EM algorithm is selected.

Another significant part of face naming is to determine the face representations in terms of facial features. One of the most common methods used to represent faces recently is based on the SIFT algorithm. Recent researches on image detection techniques have shown that local image features provide a better description on the detection of images, especially faces. One of the algorithms to extract descriptors of local features was published by David Lowe in 1999, SIFT (Scale Invariant Feature Transform) [19]. A significant reason for this method to be preferred on face detection is that it is a scale invariant algorithm, as it indicates with its name. The feature vectors extracted by the algorithm are robust to translation, rotation, scaling and illumination. Handling the face recognition problem as a face naming problem arises from the need to recognize faces on photos that are captured in uncontrolled environments. While holistic based approaches gives better results on controlled environments, feature-based approaches works better on uncontrolled environments, since they are robust to variations on images caused by environments. Hence, recent researches on image detection techniques have shown that local image features provide a better description on the detection of images, especially faces. One of the algorithms to extract descriptors of local features was published by David Lowe in 1999, SIFT (Scale Invariant Feature Transform). A significant reason for this method to be preferred on face detection is that it is a scale invariant algorithm, as it indicates with its name. The feature vectors extracted by the algorithm are robust to translation, rotation, scaling and illumination.

Although SIFT algorithm was widely applicable for 2D classifications, Bicego et al.s work [4] was one of the first attempts to apply SIFT on face classification. They use SIFT on faces using three different matching approach. The first and the simplest one is the matching pair distance, which is to compute the minimum pair

distance, DMPD, via getting the Euclidean Distance between all pairs of keypoint descriptors. The second methodology, Matching Eyes and Mouth, depends on the claim that the most significant face features are located around eyes and mouth, therefore using only those descriptors, a pair-wise matching is performed. The last methodology, Matching on a Regular Grid, improves the first two methodology, by examining their problems which are not taken into consideration, such as matching of descriptors in different locations. In the first methodology, each descriptor is matched with others, and although in the second matching criteria, eyes are matched with eyes, and mouth with mouth, still, the descriptors on the left eye are being matched with the ones on the right. Realizing that matching different descriptor locations with each other is unrealistic, in the third matching method a location dependent matching approach is applied.

Later studies on facial feature extraction with SIFT, improve the decision on locations of the SIFT descriptors to be selected from points that will be more significant on recognition. Nine keypoint descriptors, two on each corner of both eyes, two on each corner of mouth, two on the nostrils and one on the tip of the nose are selected to be the facial features. In [7], four more significant points are added, two being at the center of each eyes, one being between each eyes and one being at the center of mouth.

Another recent study on face-name association using the combination of textual and visual content is investigated by Everingham et al. [8]. This study is an expansion to [7]. Their method is divided into two sub problems, first by the alignment of subtitles and transcripts, they automatically generate time stamped character annotation, and secondly they strengthen this information by identifying if a character speaks. Finally, in order to prune the errors resulted from weak textual annotation; they include cues for face and cloth matching. The automatic alignment of subtitles and scripts, it is possible to find what is said by whom and when. This information is then combined with the visual speaker detector in order to detect the character of the speaker. They used 9 facial features of SIFT descriptors also proposed in [7]. Their face detector detects frontals, however frontal face detectors, although more reliable, will not be adequate for videos. Therefore, for the cases where a detection of face is hard, a cloth representation

is taken into account. The clothes are matched with the faces, and represented using color alone. For classification two different methods are used, one is nearest neighbor classification and the other is the SVM classification. Depending on the poses of a face the classifiers gives different results.

Another expansion to the study [7] is proposed by Sivic et al. [24]. They propose a method to label faces automatically on TV or movie materials using a weak supervision technique combined with the textual content of subtitles and script texts. Their previous work [7] on the subject suggested a method where samples faces restricted to be frontal and nearest neighbor classification was used. In this study they expanded their work to be able to detect profile faces as well, and their facial features are able to distinguish between characters. They proposed two different methods for face detection; one is for full frontal faces and the other is *3/4 view* to full left profile, and the right profiles are detected using a mirrored input image. HOG feature extraction is used along with a linear SVM classifier. Their facial features are based on 13 points on the face regarding points around mouth, nose and eyes. The results are than combined with the speaker detection they propose. Their speaker detector extracts the textual content from scripts and subtitles, if at that period of time there is a face on the screen. Their method outperforms the results where only frontal face images are of concern. As a result of this study, more faces are detected to be named.

Another face-name association study on the news videos are introduced by Le et al. [17]. They proposed a method to find important people repeatedly appearing during a certain time period in large news video databases. They divide the problem into two sub problems, one is to group similar faces in order to find dominant groups, and the other one is to label these groups. However, the problem of face-name association still preserves the same troubles, since the face images from videos, will vary in illumination, pose, hair-style, etc. Also the problem of name-face pairs not being together causes error-prone labeling. To handle these problems, for finding dominant groups, they used a relevant set correlation based clustering model which can efficiently find similar dominant groups among noisy and large groups of data. For labeling procedure, name extraction from the transcripts are performed, and the filtered extracted names matched

with their best correspondences. The results from the clustering algorithm are observed to give a ranking score in terms of being important. They decide on this rank via examining the appearance degree of the subject person that the group of images belongs to. The faces are detected using the method using LingPipe described in [2], where the names are extracted, and sorted according to their frequencies. Frequency based elimination is performed in order to remove the unimportant names. A generalized nearest-neighbor clustering method RSC (relevant-set correlation) proposed by Houle [11] is applied to find the dominant groups. The group with the higher frequency is selected as the anchor face clusters. All the anchor-faces are found and then faces belong to these clusters are removed in order to find the anchor face images not found at the first step. Using the assumption that the anchor faces appear at the similar studio settings, for the remaining set, a new clustering, based on color histograms of the already extracted faces are performed. Finally the name face association is completed using the methodology proposed by Duygulu et al. in their study [6].

Pham et al. in their study [22], also focused on name-face association; however rather than the general approach of assigning names to the faces on the images, they propose a method which aims to achieve a one-to-one assignment for names and faces. In order to do so, they proposed three significant models, one is to assume the names in the texts generate the images, the second one is to assume the faces on the images generate the names, and the last one computes the alignment of names and faces with a joint probability calculations of names and faces, P(f,n). For each image-text pair si, with Fi faces and Ni names, there are several alignment schemes a_j. In order to achieve a successful one-to-one alignment they used a standard Expectation Maximization algorithm, where their expectation step is to estimate the likelihood of each alignment a_j, for s_i; and the Maximization Step updates the probability distributions using the alignment estimations. In order to strengthen their strategy to find alignments, they use two additional scores, picturedness and namedness, to be used in P(f—n) and P(n—f) calculations. They come up with this idea based on the case that not all the names extracted from texts, or not all the faces extracted from images, have the same possibility for being significant in terms of having a corresponding face-image

just because the name is mentioned, in their own words the picturedness, or having a corresponding name just because the image contains the face, namedness. The picturedness and namedness scores are then used for the evidence of name-face co-occurance. Finally, for each story si, the alignment a_j with the highest corresponding δi_j obtained from the E-step of EM algorithm is selected.

Kumar et al., in [15] proposed a novel method for face representation. During their studies on face recognition, they realized the confusions may as well appear between, male and female, Asian and Caucasian, young and old. They proposed a method for reducing the confusions in labeling by generating a feature vector that aims to represent faces using the distinguishable facial aspects of people. They have found 65 different attributes, such as age, sex, gender, etc. for representing a face. They extract their facial features based on two methods, attribute and simile classifiers. The first method attribute classifiers, is the step where a binary classification whether those attributes exist or not is performed. The simile classifiers on the other hand, examine the resemblance of those attributes for a face to a group of reference face.

Although recent studies on the area, propose effective solutions to the problem of name-face association, the suggested algorithms works successfully for the cases where image samples are usually large in size. The popular people such as celebreties or politicians appear more-frequently on the web, therefore; collecting image samples for them is a relatively easier task compared to the task of collecting image samples for people appear less-freqeuntly on the web. Regarding these facts, in this study we focus on naming less frequently appearing people who are seen next to more-frequently appearing people on the web. The methodology will be simply to name the more frequently appearing people first, then name the less-frequently appearing ones next to them, using the textual content.

Chapter 3

Naming Multi-Face Images

With the observation that, some people, such as politicians or celebrities appear on the news related web pages more frequently, naming faces for them is a relatively easier task compared to the task of naming less frequently appearing people. In this study, using both textual and visual content, we will make use of the more frequently appearing faces on the web, to name the less frequently appearing ones. The algorithm is divided into two major steps. The first step is to name the more frequently appearing faces on the web using supervised classification algorithms. In the second step, textual content will be used for faces that are not assigned to any more-frequently appearing faces at the first step.

The overall algorithm consists of the following steps:

- Name more frequently appearing people on the Web
  - Label faces with supervised classification
  - Find the outliers (in other words the faces that are not in the list of more frequently appearing people)
- Name infrequently appearing people on the web
  - Assign names to outliers using textual content
  - Pruning the categories generated for outliers.

In the following two subsections, first the methodology used to name the more frequently appearing people and next the methodology of naming the infrequently appearing people will be described. Then a use case scenarip will be presented for a better understanding.

3.1 Naming more frequently appearing people on the web

In this section, the methods used for labeling popular faces, detecting outliers, and finally assigning names to those outliers will be explained.

3.1.1 Labeling faces with supervised classification

In this study, we label the people appearing frequently on the web by using a supervised classification setting. Although other methods proposed in the literature could also be used, this decision is made in order to get a high accuracy. We use Support Vector Machines (SVM) to train the classifiers for a number of people appearing most likely. The details will be explained in Chapter 5 briefly.

To sum up; we assign an input to the label of the closest sample in its training set. Therefore, given an image with multiple faces on it, the supervised classification method SVM matches the faces on that image with the closest category from our dataset of more frequently known people.

For example, Figure 3.1 illustrates the classification for an input image with two faces. Each face on the image is compared to the face categories in the dataset. In this step of the algorithm, each face image will be assigned to the name of its closest category found by the classification algorithm. To go further; we need to distinguish the faces that do not belong to any category. The next section will introduce the outlier detection, in other words, the detection of the less-known faces on the images. In the rest of the thesis, we will refer to the

images with multiple faces on it, as *multi-face* images.

3.1.2 Finding the outliers

In the previous step, each face is labeled with the category having highest confidence value. After finding out the highest confidence values for each face, a threshold value is calculated in order to decide whether the face should really be assigned to a category or should be labeled as an outlier. This means that the faces left as outlier in this step do not belong to any category in the list of more frequently appearing people, but likely to be one of the less frequently appearing people. (see Figure 3.1(c)).

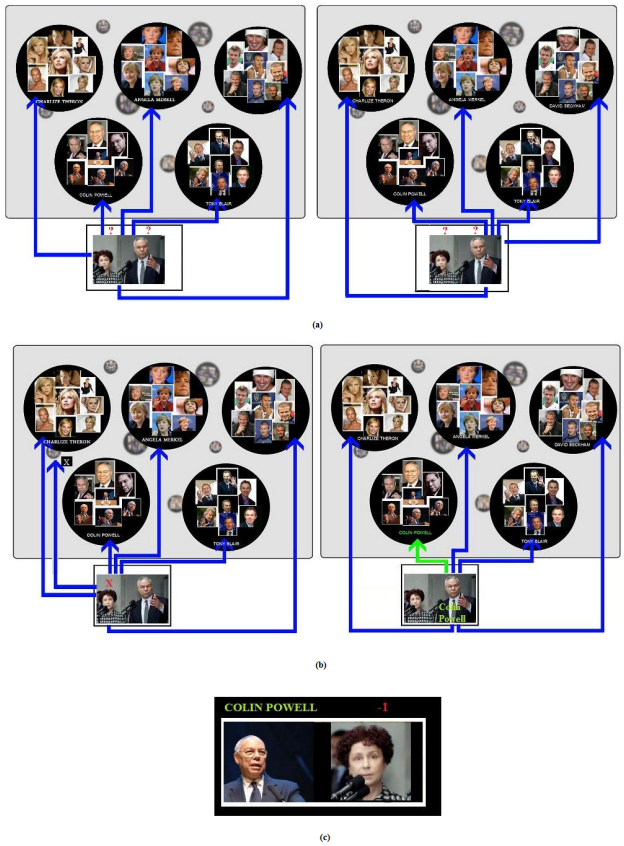

Figure 3.1: (a) Classification without any outlier detection. (b) Labeling without any outlier detection. (c) Labeled faces with outlier detection. "-1" means the face image is an outlier.

3.2 Naming infrequently appearing people on the web

In this section, the methods to name less frequently appearing people will be described.

3.2.1 Assigning names to outliers using textual content

The textual content associated with the images including the face of a person is likely to include the name of that person. However; there may be several names in the text, several faces on the images, and with any of the correspondances also missing (see Figure 3.2). With the supervised classification method described in the previous section, the name-face association for frequently appearing people becomes more reliable. The outliers detected in the previous step are now labeled using the textual content with the names of frequently appearing people are excluded.

Figure 3.2: Sample news images and their textual contents provided by FW Dataset.

To understand how labeling works, let's consider the example in Figure 3.3 where Colin Powell and Ana Palacio appear together. Besides the names of the two people in the list of more-frequently appearing people. With the steps explained in the previous sections, Colin Powell is correctly labeled however, since the person next to him is not a member of the most frequently appearing people list, she will not be recognized by the system. Therefore, she will be labeled as an outlier. In this step, the less frequently appearing people will be named with the extracted names, which are not yet assigned to a face. For this sample, the extracted names will be *Colin Powell*, *Ana Palacio*, *State Department in Washington* and *Spain*. Although the last two proper names, does not belong to

people, they will also be extracted by the system.

Figure 3.3: Input image of Colin Powell and a less known person (Ana Palacio). Colin Powell is correctly labeled; and Ana Palacio is labeled as outlier. The names are extracted from the textual content.

Examining the faces labeled on an image, the names that are already matched with a face will be eliminated from the list of names extracted for the image. In this case, Colin Powell is already labeled by the classification system; therefore, the less known person on this image will be assigned with three of the names extracted from the text; *Ana Palacio*, *State Department in Washington* and *Spain*. For each extracted name, bunch of face images will be collected from other image-text pairs. Correct face-name association by eliminating the irrelevant faces from this collection will be explained in the next section.

3.2.2 Pruning the categories generated for outliers

As explained in the previous section, the names extracted from the textual content are assigned to the outlier faces on the corresponding images.

This step results in collections where names are associated with a set of faces; however, there are faces belonging to the name just as there are faces that are

Figure 3.4: Less frequent face image collection for extracted name Ana Palacio.

irrelevant (see Figure 3.4). As expected with the intuition of the general situation that a face of a person appears on the web around his/her name, the number of relevant images in the collection is more than the number of irrelevant images. With this assumption, it becomes possible to prune the irrelevant images from the images belonging to the name.

The irrelevant images are pruned by the following two steps:

- Generate a dissimilarity graph for the images collected for an outlier name
- Find the most similar images in the collection in this graph

3.2.3 Dissimilarity graph construction for outliers

In this step, a dissimilarity graph, where the nodes represent images, and the edges represent dissimilarity weights between the nodes, will be constructed for face images assigned to an outlier name. Let there are n numbers of face images matched with the name N. The graph G(V,E), where V is the set of face images, and E is the set of edges between them, will be an nxn matrix. E_{ij}, the edge between the i^{th} node and the j^{th} node, is the dissimilarity weight between the face image I_i and I_j. Let the feature vector for a face image be *Fv(I)*, the dissimilarity,

D, between face image i, I_i; and face image j, I_j is the Euclidean distance between the feature vectors of these faces;

$$D(Ii, Ij) = Euclidean(Fv(Ii), Fv(Ij)) \tag{3.1}$$

$$D(Ii, Ij) = \sqrt{(Fv(I$$

$_i)^2$ - Fv($I_j)^2$) (3.2)

Having constructed the dissimilarity graph for an outlier name, the most similar faces in the graph will be found in order to eliminate the faces that do not belong to the specified name (see Figure 3.5).

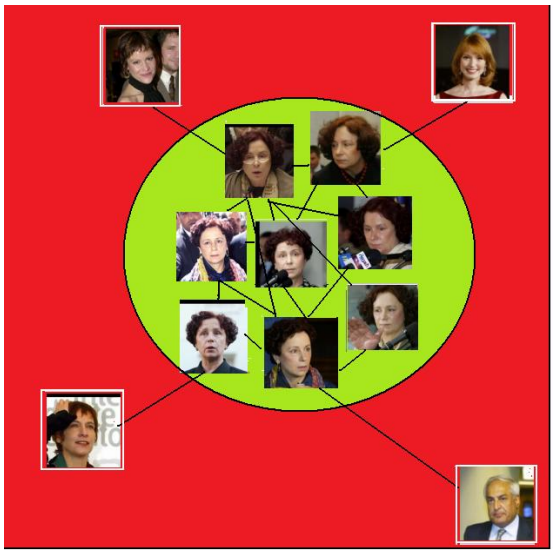

Figure 3.5: Constructed graph of the images collected for extracted name Ana Palacio.

Although there are different graph based methods on finding the densest subset in a graph, those algorithms give successful results for larger sample image sizes [21], [10]. However, in our case, the sample face images collected for less-frequently appearing names are limited, therefore we have proposed a different

algorithm. In order to find the most similar face images, we apply an algorithm inspired by Borda Rank Algorithm [1] on our dissimilarity graph. (see Algorithm 1)

Algorithm 1 Ranking of images : Finding the most similar face in a collection of face image

1: n : no Of Images collected for a name
2: $sortedD$ = sort the dissimilarity matrix D(Ii,Ij) (an nxn matrix)
3: $sortedDInd$ = the indices of the sorted matrix D (an nxn matrix)
4: **for** each image i **do**
5: **for** each value j of the row D **do**
6: $rankOfImgi$ = find the rank of the image i at the jth row of sortedDInd
7: $rankD(1, i) = rankD(1, i) + rankOfImgi$
8: **end for**
9: **end for**
10: $rankD = rankD$ / n
11: {rankD is a 1xn row vector where rankD(1,i) is the total dissimilarity rank of image i for among all n images.}

For an outlier name, a *1xn* row vector (*rankD* in Algorithm 1), is constructed to represent the outlier face. At each index of this row vector, a value for the corresponding face image is hold. This value is the total rank of the image in terms of dissimilarity among the other images. Since D is a dissimilarity matrix, when it is sorted in ascending order, the first indices of the i^{th} row will carry the most similar images to itself. As we go to the last indices, the less similar faces to face image i will be found. Therefore, an i^{th} row of *sortedDInd* will carry the most similar faces to face image i in ascending order. With the help of *sortedDInd*, one can find the rank of all images, in other words, which images are mostly seen at the first indices. So the i^{th} index of *rankD* holds the total rank count of i^{th} image among other images. The lower this value means the image is strongly similar to the majority of the image collection. As a result of this algorithm, among the collected less-known face images for an extracted name, the most similar images, in other words, the relevant images among this collection will be found. (see Figure 3.6)

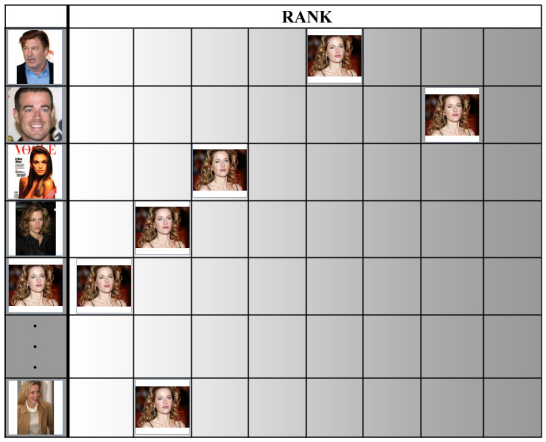

Figure 3.6: Ranking algorithm for images collected for a less-frequently appearing face. The images with higher ranks will be the outliers in the collection.

3.3 A use case scenario

For a better understanding of our methodology, this section will present a use case scenario for explaining each step of our approach given in the previous sections.

3.3.1 Name more frequently appearing people on the web

3.3.1.1 Labeling faces with supervised classification

We construct a dataset with a set of names to be the popular (more-frequently appearing) faces, and another set of names is selected to be the names of less-known (less frequently appearing) faces. With the intention of simulating a multi-face web image, one face from each set is taken. Additionally, knowing that a face image is encountered when its name is mentioned in its textual content, we randomly generate a corresponding textual content which may or may not contain the names of the faces on that image.

First each face on an image is labeled with one of the categories in our list of frequently appearing people. Figure 3.7 illustrates a set of images generated

to be a two-face image where the person on the left is popular, and the person on the right is a less-known person. Therefore, in this figure while *Dave Chappelle*,*Donald Trump*, *David Beckham* and *Adam Sandler* are selected to be the popular faces on the web, the faces on the right side of the images are selected from the names that are chosen to be the less-known faces. As it is mentioned before, in this very first step, each face on the image will be labeled by the classifier, with a name from the categories of popular names in our dataset.

Figure 3.7: Labeled faces without any outlier detection.

3.3.1.2 Find The outliers

The labels assigned to each face are actually the names of the faces which are found to be the best match for an input face image. In other words, although the faces on the right are labeled with a name, actually they are only labeled with the name of the face that the algorithm finds as a best match. However, examining through the confidence values of a face being in its matching category,

we find out that the probabilistic results for less-known faces are lower, compared to the faces labeled correctly with the categories of popular ones. Therefore, as it will be explained in Chapter 5, a threshold value is calculated for detecting the less-known face images. Figure 3.8 illustrates the labeling results for the images given in Figure 3.7 after using a threshold value for labeling process which results in labeling less frequently people as outliers (shown -1)

Figure 3.8: Labeled faces with outlier detection.

3.3.2 Name less frequently appearing people on images

3.3.2.1 Assign names to outliers using textual content

In Figure 3.7, the first image contains the faces of *Dave Chappelle* and *Gillian Anderson*. The random selection of popular and less-known name category sets in that case resulted as *David Chappelle* being the popular person on that image, and *Gillian Anderson* being the less-known person. Hence, the face of *Gillian Anderson*, who is labeled as an outlier in the previous section, will be labeled in this step, with a name from the textual content.

The textual content is also randomly generated for both faces on the image,

with a higher probability for containing correct names in the text. But there is also a low probability of having an irrelevant name, and not even having any correct name in the entire text. At this step, name generation and extraction for images are simulated. Below, in Figure 3.9, results for the randomly generated image of *Dave Chappelle* and *Gillian Anderson*, and its corresponding textual content is explored. The extracted names are found as *Dave Chappelle*, *Gillian Anderson* and *Gael Garcia Bernal*.

Figure 3.9: An input image of Dave Chappelle(as celebrity) and Gillian Anderson(less frequently appearing face) is labeled correctly for Dave Chappelle, and the image of Gillian Anderson is correctly detected as outlier. Then the names are extracted from its textual content.

Although, both the names *Dave Chappelle* and *Gillian Anderson* are related to the image, *Gael Garcia Bernal* is mentioned without the appearance of his face. With the detection of the outlier face and the extraction of the names from text, the outlier face will be matched with the names extracted. In order to limit the extracted names for the assignment of less known faces, any names that are already assigned to a face on the same image are eliminated (Figure 3.10).

The remaining names on the text will be candidate names for the less frequently known person. Examining through the names extracted for this image, *Dave Chappelle* is eliminated from the list, since it is already used to label the face on the left. Therefore; the outlier face on the right is matched with two names; *Gillian Anderson* and *Gael Garcia Bernal*. As the operation proceeds, more and more faces that are labeled as outliers at the first step will match with the extracted names (see Figure 3.11).

Figure 3.10: Name matching for Gillian Anderson among the names extracted from the textual content of the input image of Dave Chappelle(as celebrity) and Gillian Anderson(less frequently appearing face).

To sum up; any face image belonging to "Gillian Anderson" and detected as outlier, will be matched with the name "Gillian Anderson" if the textual content contains her name. As a result, there will be a bunch of face images collected for "Gillian Anderson". However, there will be irrelevant images as well, since the matching procedure only depends on the textual contents, which might not contain the correct names. In other words; although *Gael Garcia Bernal* himself does not exist in the image, his name is mentioned in this simulation. As a result of this, the face of Gillian Anderson is also matched with the name Gael Garcia Bernal. However, irrelevant images collected for a name will be less than the relevant ones. Therefore, searching for the most similar images in a set of face images will lead us to find the correct face for the extracted name.

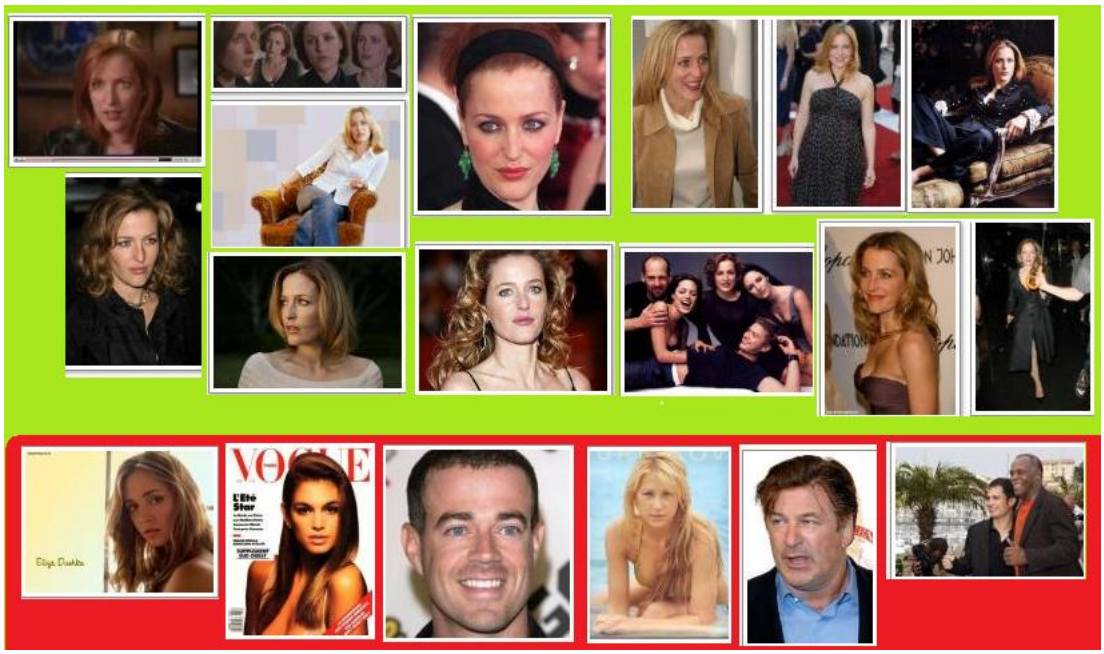

Figure 3.11: Face images collected for the extracted name “Gillian Anderson”.

Chapter 4

Dataset and Facial Features

4.1 Dataset

The algorithm has been applied on two different dataset. The first dataset is a subset of the dataset Labeled Faces in the Wild, collected by Berg et al.[3]. The entire dataset consists of 31.280 images from 1.249 categories of people. The images are collected from Yahoo! News over a two years period by Berg et al. The dataset contains the original news images, their captions and the cut-out face images for each face detected on these news photos.

The second dataset is a subset of PubFig dataset, which is collected from the web by Kumar et al. Their evaluation dataset consists of 42,879 images for 140 categories of names. For each face, they specify 65 different features some of which are the attributes of being male, Asian, white, black, baby, child, youth, etc. 65 facial features are given in A.2.

Faces on both datasets are considered to be more realistic compared to the existing face recognition dataset images.

The collected images are captured *in the wild* rather than controlled environments, therefore, the faces in these photos vary in pose, illumination and

expressions; even they are exposed to occlusions.

Faces in the Wild dataset already consists of multi-face images and their corresponding textual contents. However, in the PubFig dataset, we only have images of single faces. In order to apply our algorithm on this dataset, images with multiple faces, and their textual contents are required. Therefore, we have used this dataset to form a simulation of web photos collection. Examining the Berg's dataset, we find out that the content belongs to an image may or may not contain the correct names of the faces on that image. However, having the correct name in the text is more likely than not having it. On the other hand, the name extractors not only detect the names of the people but also the location names and proper names are detected. Hence, in order to simulate a realistic content for an image, we generate names in following ways. If we want an image to contain n faces, we generate (n+1) different name slots, the first n slots belonging to the n faces respectively, and the last slot belonging to any location or proper name. However, for the simulation to be realistic, an error rate of having an irrelevant name in the text should be taken into account. In the light of the information that textual contents have a tendency to contain accurate names more frequently, after a series of empirical experiments, we decided to give 80% probability of having an accurate name, where 20% of the time the name will be any other name than the ones that belong to the faces in that image. Meanwhile, the contents may not always contain a location or proper name which could be detected as a person name; therefore there is a 50% probability of that last slot to exist. If it exists, a random name from the combination of all the category names except the ones in that image, and the formerly constructed bunch of location names will be selected for that slot. To clarify our method, let there are n faces on an image, belonging person A, B, ... ,and X. The textual content belonging to this image will generate names as illustrated in Figure 4.1.

In our study, we have used subsets of the two dataset Faces in the Wild, which will be referred to as FW, and Public Figures Face Database, which will be referred to as PubFig in the remaining of the thesis. The following section will give brief information on the facial features extracted from detected faces.

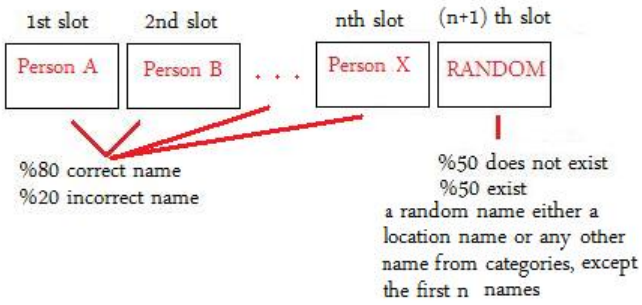

Figure 4.1: Random name generation for textual content of an image.

4.2 Facial Features

Recent studies on face recognition applied on the LFW [12] dataset support that, it is difficult to achieve a high accuracy rate of labeling faces for images that are captured under uncontrolled environment. Not only illumination, pose, focus resolution of images cause bad labeling in face recognition systems, but also make-up, hairstyle, eye-classes, facial hairs (beard, mustache) and several other person and environment related changes on the face, make recognition difficult for researchers of the area. In order to reduce the effects of such problems, feature vector selections on face representations are important. Two different methods for face representations are applied for our study. One of them is PubFig Representation which is a novel approach introduced by Kumar et al. [15], and the other one is the SIFT [19] descriptors extracted for 9 specific facial points, as its name indicates a scale-invariant method robust to pose, illumination, scale and etc.

4.2.1 PubFig Facial Features

When the incorrectly labeled data is examined by Kumar et al,[15], it is realized that the confusion in labeling may as well appear between man-woman, young-old, Asian-Caucasian to a great extend. They claim to reduce this confusion

by suggesting a novel method for face verification of images captured under uncontrolled environment. Using the common idea of extraction and comparison of high-level visual features they explore their contribution under two methods, attribute and simile classifiers. They mention that it is easier to collect data for their face recognition system, since their visual features are robust to pose, illumination, expressions etc. Contrary to existing methods in face recognition [26], their method does not require a pre-alignment of image pairs and through this contribution they get rid of a computationally expensive work. Although they use visual features common to existing methods, their features are different in terms of representation of faces. In their own words, their visual features "*provide information about the identity of an individual*". They suggest two methods for their novel visual features; one is the attribute classifiers; which is to recognize the describable attributes; and the other is the simile classifiers for recognizing the similarity of those attributes to a set of reference people. In our study, we have used their attribute clasifiers, the details of their attribute classifiers are explained below.

4.2.1.1 Attribute Classifiers

Inspired from the name of the method described in [15] this method is called attribute classifiers based on the idea of extracting the facial attributes of individuals. In this step a binary classifier is trained for recognizing whether those *describable aspects of visual appearance* (such as age, hair color, race, sex etc.) exist or do not exist. Up to now they have built 65 attribute classifiers, explored in A.2, and via these attributes they can recognize faces despite the illumination, pose or expressions. As it is clarified in figure 4.2 most of the attribute classifiers belong to the two face image of Hale Berry is strongly close to each other, despite the differences in pose or illumination.

In order to construct their attribute classifiers, first the low level features are extracted, then the visual traits are computed.

1- Extract Low Level Features

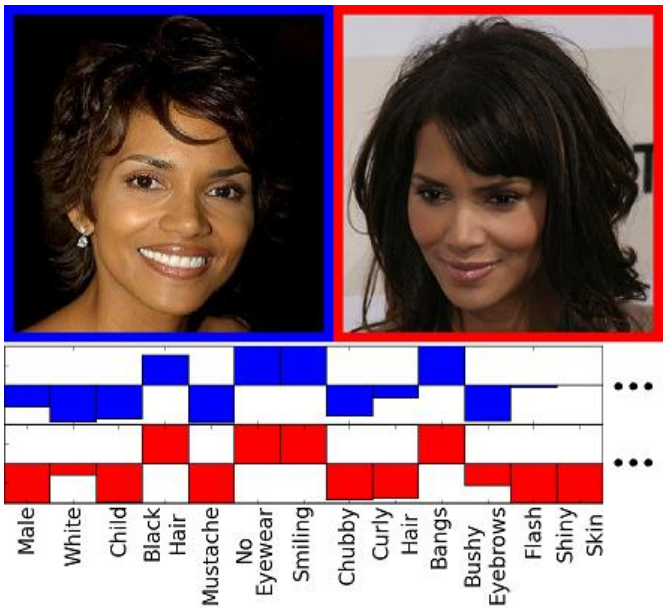

Figure 4.2: Attribute Classifiers agreement despite the differences in pose, illumination, etc. (this image is taken from [15]).

For every face image *I*, a feature vector *F(I)* is constructed. *F(I)* is simply constructed by the concatenation of *k low level feature vectors* f_i *= 1, ..., k.* The low level features are constructed via extracting the image intensities in RGB, HSV color spaces, edge magnitudes and gradient directions of fiducial point locations on the regions manually selected from the rectified image of the detected face region outputs of commercial face detectors.

2- Compute Visual Traits

In this step the attribute classifiers in other words the trait vector *C(I)*, *n* trait classifiers $C_1 ... C_n$, is computed using the extracted feature vectors. C(I) = $< C_1F(I), ..., C_nF(I) >$

Based on their assumptions, face verification systems may as well confuse people of different sex, gender, age and etc. Kumar et al.[15] comes with a novel solution to the confusion problem. They propose a method where the facial features are extracted based on people's different attributes that can distinguish

them from other people, and evaluate those attributes via comparing them with a set of reference people. Their experiments proved that, this first and novel method applied on face verification, results in a lower error rate in confusion compared to LFW [12] . (31.68% to 23.92%)

4.2.2 SIFT Descriptors

As a second approach for feature vector selection, SIFT features are extracted for 9 facial features mentioned in Everingham et al. [7]. The selected 9 facial features are chosen to be robust to translation, illumination, pose, etc. Therefore; having a face image collection captured under uncontrolled environment as our dataset, extracted SIFT descriptors will be a strong way for representation of faces. Those specific 9 facial points explored in Figure [4.3], are the left and right corners of each eye, the two nostrils and the tip of the nose, and the left and right corners of the mouth.

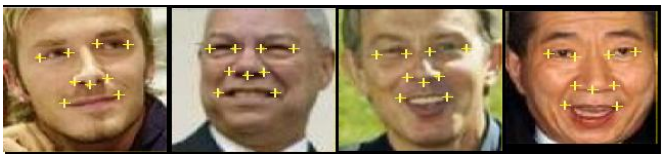

Figure 4.3: Specific 9 Facial Points.

For each 9 point, a 128x1 column vector of SIFT descriptors are extracted. As a result, a face image is represented with a 128x9 matrix.

Chapter 5

Experiments

In this chapter, we will evaluate the results of our algorithm and explore the accuracy rates of each step.

5.1 Construction of the dataset

As introduced in section 4.1, two different sets of data, Faces in the Wild, and Public Face Figures Dataset, are used in order to form our dataset. For the rest of this study we will refer to Faces in the Wild Dataset as FW, and Public Face Figures Dataset as PubFig. The FW dataset already consists of multi-face images (images with multiple faces on it) and their corresponding contents. However, in the PubFig dataset, we only have images of single faces. Therefore, in order to apply our method, while FW Dataset does not require any additional work; slight changes and additions are necessary for PubFig. Let us first give a brief information on the additional work applied on PubFig Dataset.

In the following two sections; using PubFig Dataset, first, the simulation implemented for multi-face image generation and then random generation of textual contents belonging to corresponding multi-face images will be explained.

5.1.1 Multi-face image generation

There are approximately around 300 face images for each 140 categories of names in Public Figures Face Database (PubFig). Among all these 140 categories, the category with the maximum number of image collection contains 1536 images, while the one with the minimum number contains only 63 images.

A total of 42.879 images for 140 people are collected for PubFig Dataset generation by Kumar et al. Since the subject of this study is to name less frequently appearing faces using frequently appearing ones; we needed to construct two sets of names among those 140 categories, for both most frequently and less frequently appearing people on the web. Therefore, we randomly select some of the names to be the faces of most frequently appearing people, and some of them to be the less frequently appearing ones.

We choose to have n categories of names, where k of them belong to popular faces and l of them belong to less frequently appearing faces. Using 10-fold cross validation, at each step k random names for popular faces are selected among 140 names, and for less frequently appearing faces, l random names are selected from the remaining 140-k names. After determining the names for less-frequently and most-frequently appearing people, w random face images for both name sets (a total of 2*w face images) are selected from the category sets of images belonging to these randomly generated names. In order to generate an image with two faces, one from each w random face images are put together. To make the evaluation easier, while the faces placed on the left sides of the image are selected from the categories of popular names, the faces placed on the right sides are selected from the categories of less-known people names (Figure[5.1]). n have been empirically changed during our experimental work.

5.1.2 Random generation of textual content

Considering the fact that, for a query name, image search engine results are pretty satisfactory, we can arrive at the conclusion that generally textual contents around

Figure 5.1: Sample generated two-face images. Faces on the left belong to popular people, faces on the right belong to less-known people.

face images contain the names of the corresponding faces. However; it is possible to retrieve images with irrelevant names in its textual content. For a query name "George W. Bush", images of "Saddam Hussein" may appear in the result set. Therefore; in our experiments, in order to simulate a realistic textual content generation we include some error rate in having correct names for face images.

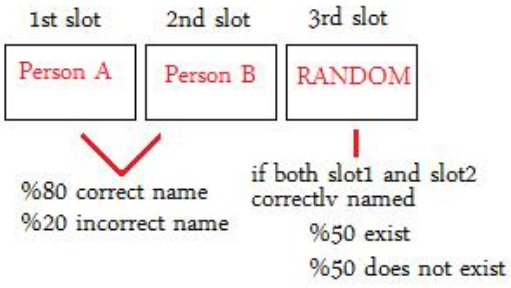

Figure 5.2: Random name generation for textual content of an image with two face.

For images with two faces, textual contents are generated as illustrated in Figure 5.2. For an image withtwo faces on, we allocated three slots for name generation. The first slot belongs to person A, the second slot belongs to person B, and the last slot is a random name. However, as explained before, we will

consider some error rates. Let L be the set of names in our dataset, and R be the set of random names, either belonging to a location name or any name from set L except persons A and B. The first and second slots will have the correct name for persons A and B respectively, with a probability of 80%. The third slot will exist with a probability of 50% only if both of the slots are correctly named. This way, for the random text generation to be realistic, we keep a high possibility of having either a location name or an irrelevant person name in the textual content. The probability values are selected empirically. Results from different probability values are evaluated during our experiments and will be explored in the later sections.

To clarify the text generation procedure, a possible execution for an image with persons A and B who are in the Engineering Building is given in Figure 5.3. The figure illustrates the simulation of the text generation for that image where the two slots are correctly labeled for Person A and B, and there exists an a location name as an irrelevant name.

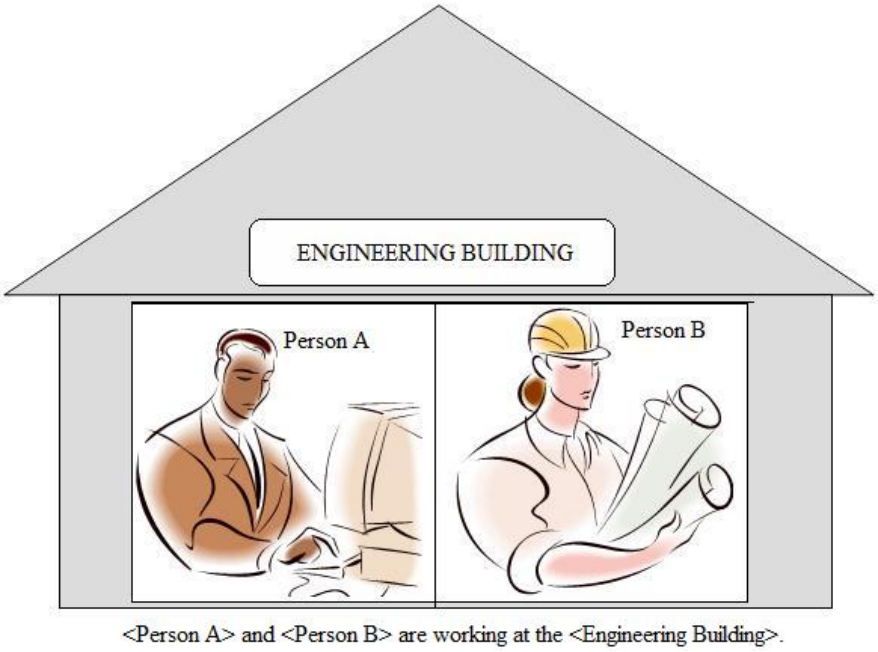

Figure 5.3: Correct Text Generation Including a Location Name.

5.2 Evaluation criteria

In order to evaluate our results, we used precision and recall evaluation criteria based on *true positive (TP)*, *true negative (TN)*, *false positive (FP)* and *false negative (FN)* results. For a full comprehension of precision/recall values, let us first introduce the TP, TN, FP and FN. Each value is separately calculated for both labeling of popular faces and less-known faces.

True Positives (TP); are the number of correctly labeled faces of a category.

True Negatives (TN); are the number of faces which are correctly assigned as non-query face. In our case, for more frequent and less-frequent face labeling, true negatives will be the number of faces, correctly labeled as outliers.

False Positives (FP); are the number of faces which are incorrectly assigned to the desired category.

False Negatives (FN); are the number of faces which are incorrectly assigned as not belonging to the desired category.

We will refer to TP, TN, FP, FN for outliers as TP_outliers, TN_outliers, FP_outliers, FN_outliers and for most frequently appearing people as TP_famous, TN_famous, FP_famous, FN_famous, respectively.

Let there be N faces in our multi-face image collection that contains faces from different categories of popular people and the outliers, namely the faces that do not belong to any of the categories included in our face-image database. Let N_i represent the number of faces belonging to the category i (i.e ground truth for category i), and O_i be the number of outliers. In our study we need to evaluate our success rate in terms of two different concerns; one is the accuracy of assigning faces to their corresponding categories; and the other is to be able to detect a face as an outlier, if it truly does not belong to any of the existing categories in the database.

To sum up, in our algorithm, two different approaches for SVM are applied

using the modified SVM, LibSVM. As introduced in section 5.3, while one-class SVM, labels the faces as a person, or non-person query; multi-class SVM assigns each face to a label from the existing categories. Hence in one-class SVM any other face that is not labeled as *person* is an outlier, on the other hand, in multi-class SVM, we need a threshold value to label a face as an outlier. After these steps, the faces will be labeled either as a popular face, or an outlier; therefore, we will evaluate our results of precision and recall values both for popular(famous) person labeling and outlier detection. Knowing what TP, TN, FP, FN refers to, the precision and recall values will be;

Precision (PR) ; is the ratio of number of correct results found by the system to the number of results found by the system

Recall (REC); is the ratio of number of correct results found by the system to the number of correct results

We will separately calculate these values for popular faces labeled and outliers detected.

$$PR\_Outlier = \frac{TP\_Outlier}{TP\_Outlier + FP\_Outlier} \tag{5.1}$$

$$REC\_Outlier = \frac{TP\_Outlier}{TP\_Outlier + FN\_Outlier} \tag{5.2}$$

$$PR\_Famous = \frac{TP\_Famous}{TP\_Famous + FP\_Famous} \tag{5.3}$$

$$REC\_Famous = \frac{TP\_Famous}{TP\_Famous + FN\_Famous} \tag{5.4}$$

For the second step of our algorithm, *Naming Less Frequent Faces*, we will use the same criteria; however, in this step PR_Outlier will refer to irrelevant face images matched with a name of a less-frequently appearing face. On the other hand, we will use new variables, *PR_QueryImage* and *REC_QueryImage*, for evaluating the accuracy rate of finding relevant face images collected under the names of less-frequently appearing faces.

$$PR\_QueryImage = \frac{TP\_QueryImage}{TP\_QueryImage + FP\_QueryImage} \tag{5.5}$$

$$REC\_QueryImage = \frac{TP\_QueryImage}{TP\_QueryImage + FN\_QueryImage} \tag{5.6}$$

The following three sections will give experimental results on the two datasets, PubFig and FW, and finally we will compare the results from both datasets. But first, let us give brief information on our classification method. As already introduced in Chapter 3, in order to achieve high accuracy in correct labeling of the more frequently known faces, we choose to use a modified version of a commonly preferred supervised classification method SVM.

5.3 Support vector machines

Support Vector machines are one of the most preferred supervised classification methods in machine learning. It is a binary classification algorithm to predict a given sample to belong one of the two categories. Given a training data of two different categories, SVM training algorithm, builds a model to determine which one of the two classes the desired sample belongs to. To simply explain, the output of the model provided by SVM is the representation of the training samples as points in space modeled by the feature vector of each training sample. The classification algorithm, predicts to which one of the area does the new point map to.

In a more formal explanation, in order to classify the samples on a high dimensional space, SVM generates a hyperplane which divides the points of the corresponding samples into separate areas.

Although SVM is a binary classifier in its origin, it is possible to use this algorithm for multi-class classification. The multi-class problem can simply be solved by viewing the problem as a multiple binary-class problem. The classification of multi-class via binary classifier SVM is either computed by the classification of

one-class versus all the others, or by a pair-wise classification in-between each pair of classes.

In our case, a probability based multi classification with SVM is executed, using the open source codes provided in LibSVM [5]. As indicated above, LibSVM generates a model as the output of its trainer. The details of the optimization problems used for training the models are explained in LibSVM[5]. The inputs for their *svmtrain* function are; an ntxm matrix for the training data, an ntx*1* column vector for the category labels of the corresponding training data, and some parameters, which will be explained below, for SVM to train the model as desired. nt is the number of training data sample ($nt = nt_i * k$, where k is the number of classes, and nt_i is the number of training data sample for i^{th} class) and m is the dimension of the corresponding feature vector.

The usage of *svmtrain* function will be as follows;

```
model = svmtrain(trainingClasses, trainingData, '-c 1 -g 0.0154 -b 1');
```

Parameters in *svmtrain* function are explained in LibSVM as follows;

- c cost: set the parameter C of C-SVC, epsilon-SVR, and nu-SVR (default 1)
- g gamma: set gamma in kernel function (default 1/num_features)
- b probability_estimates: whether to train a SVC or SVR model for probability estimates, 0 or 1 (default 0)

After training the model, classification step will take place. The classification is performed by *svmpredict* function of LibSVM. Inspired from the method used by Friedman, 1996 [9] and Kreβel, 1999 [14] , LibSVM uses *one-against-one* approach for multi-class classification problem. Let k be the number of classes, they construct *k(k-1)/2* classifiers, and each classifier trains the data by two different classes. Each data point is then voted for each binary classification and

the class with maximum vote is assigned to that data point. If there is more than one maximum value for a point, although they admit that it is not a very good strategy, they choose to select the one with the smaller index.

Finally the voting is performed for each point to be assigned to a class. The usage of the function *svmpredict* is as follows;

```
[predict_labels, accuracy, prob_estimates] =
                    svmpredict(TestClasses, TestData, model, '-b 1');
```

The parameter '-b 1' indicates that the probability estimates will be calculated. The outputs of the svmpredict are; *predict_labels*, *accuracy*, and *prob_estimates*;

prob_estimates: an ntxk vector of confidence values (the probability estimates) for each face to be in any of the k categories.

predict_labels : an ntx*1*column vector of labels predicted by *svmpredict*. This is the category with the highest *prob_estimates* value for a face. In other words, the label of the i^{th} face is the index of the maximum value in the i^{th} row of *prob_estimates*.

accuracy : the accuracy rate of successfully labeled data.

$$accuracy = \frac{Number\,of\,correctly\,labeled\,data}{Number\,of\,total\,data} \tag{5.7}$$

5.4 Experimental results on PubFig dataset

PubFig is our second dataset to apply our methodology. The first one is FW. However, since we were able to complete all the steps of our algorithm with PubFig dataset, we will first give experimental results for it. PubFig dataset, provides feature vectors for 140 categories of names, including a total of 42,879

face images. As explained in section 5.1.1 images with two-faces are generated, and on these images one face belongs to a popular person, while the other belongs to a less-known person. Both popular and less-known names are selected from 140 categories of names. In order to evaluate the success of our algorithm we choose different values for variables mentioned in section 5.1.1. To get an average rate of success, 10-fold cross validation is applied. At each step of 10-fold cross validation, we have selected n names ($50 <= n <= 100$) among 140 categories, which will include both the popular and less-known name sets. We decided to have a ratio of 60% to 40% between the number of categories selected, respectively, for popular and less-known name sets. For SVM classification, we chose different sizes of training data for evaluating the effects of training data size, TS, on our algorithm. We have selected five different training sizes which are; 20, 30, 40, 50, and *half size of the total data* (which will be referred to as *halfSize*) for each category.

Further in our experiments; we chose n to be 50 and TS to be *halfSize* (the half size of the data provided for each category). Therefore, for n=50 we choose to have 30 categories for the set of popular names and 20 categories for the set of less-known people names. For each set, 500 images are randomly selected. As a result of this selection, having 500 face images for 30 categories of popular names, and 500 face images for 20 categories of less-known names, multi-face images will be generated. The number of categories for popular people will be greater than the less-known people, as it is the case on the web images. And selecting 500 images for 20 categories of less-known faces will result in having more number of face image samples per less-known face category compared to the number of face image samples for each popular name categories. Having one face image from each set, we will form 500 images with two-faces. Figure [5.4] illustrates the two-face image generation procedure. In the following sections, the experimental results for variations in the size of n and TS will be evaluated, and following this section, the algorithm steps for the selected variables will be explored.

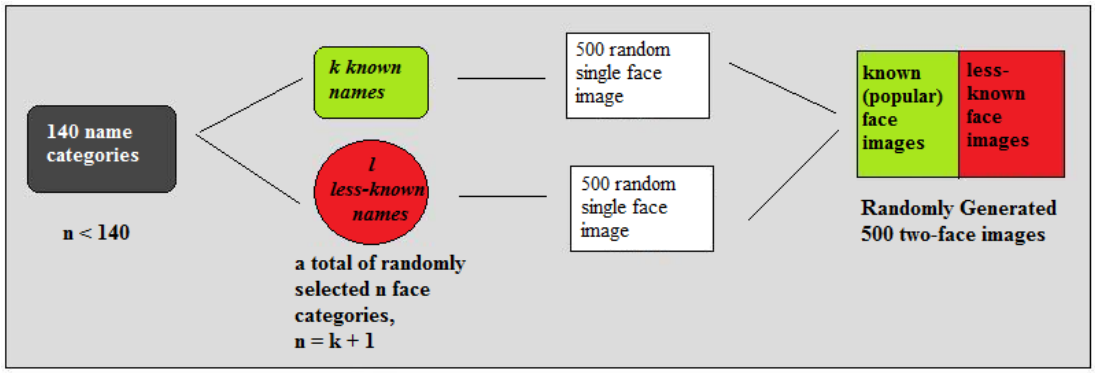

Figure 5.4: Random famous and less-known face combination for two-face image generation.

5.4.1 Experimental results for different values of n, TS and probability of correct text generation

In this section, we will give the results of our experiments for naming frequently and less-frequently appearing faces on different sizes of n (the number of categories selected both for less-known and popular names), and TS (the training size) and $\text{Prob}_{C}orr(theprobabilityofcorrecttextgeneration)$

5.4.1.1 Experimental results for different values of TS

Naming Most Frequently Appearing People on the Web

In the first step of our algorithm, the accuracy rate for labeling the most-frequent faces and the outlier detection will be explored. As it is explained before, we have generated 500 images with two faces, one face belonging to a popular person, and the other face belonging to a less-known person, in other words, to an outlier. Since SVM classification is a supervised algorithm, we need to have a training data for the categories of popular faces. In this experiment, selecting n=50, we decided on the training data size, TS. During our experiments TS is selected to be TS = { 20,30,40,50, $halfSize$(Half Of The Data Size for each Category) }. We

have selected n, the total number of popular and less-known name categories, as 50 (n=50) in this step, for the execution not to be time consuming. In outlier detection, the second threshold value which will be explained in 5.5.1.2 is used.

k being the number of popular name categories, and l being the number of less-known name categories, for 10-fold cross validation with k=30 and l=20 randomly selected names, the precision and recall values of popular face labeling and outlier detection on different values of TS = { 20, 30, 40, 50, *halfSize*} is introduced in Table [5.1]

Table 5.1: Average Evaluation Results For Different TS Values

TS	PR_Outlier	PR_Famous	Rec_Outlier	Rec_Famous
20	59.9606	31.6107	84.62	36.44
30	63.5517	39.4743	83.9	45.78
40	65.1252	43.2857	83.16	50.56
50	67.6669	48.2822	83	56.44
halfSize	77.6375	62.5487	80.36	74.8

Figure [5.5] illustrates the average precision and recall values of 10-fold cross validation, varying for different training sizes for labeling of most-frequently appearing faces. The precision and recall for popular face labeling increases, as the training size, TS, increases. Since SVM is a supervised classification, less training size gives low accuracy rates. However, after a while the rate of change in the increase of accuracy is not that significant, except the last TS value, which gives excessively higher results. For some categories there are over a thousand face image samples; therefore, giving half of the data as training size significantly outperforms the results when TS = 50, for such cases. As indicated in this figure the average precision value increased from 31% up to 48% for 20 <= TS <= 50. The recall value on the other hand, shows a higher increase, from 36% to 56%. When we take TS as the half size of the data, the recall value becomes 74% and the precision value becomes 62%. As a result, as the TS increases, the number of faces correctly labeled increase to a great extent. Consequently, the number of false positives decreases, as well (see Figure [5.6]).

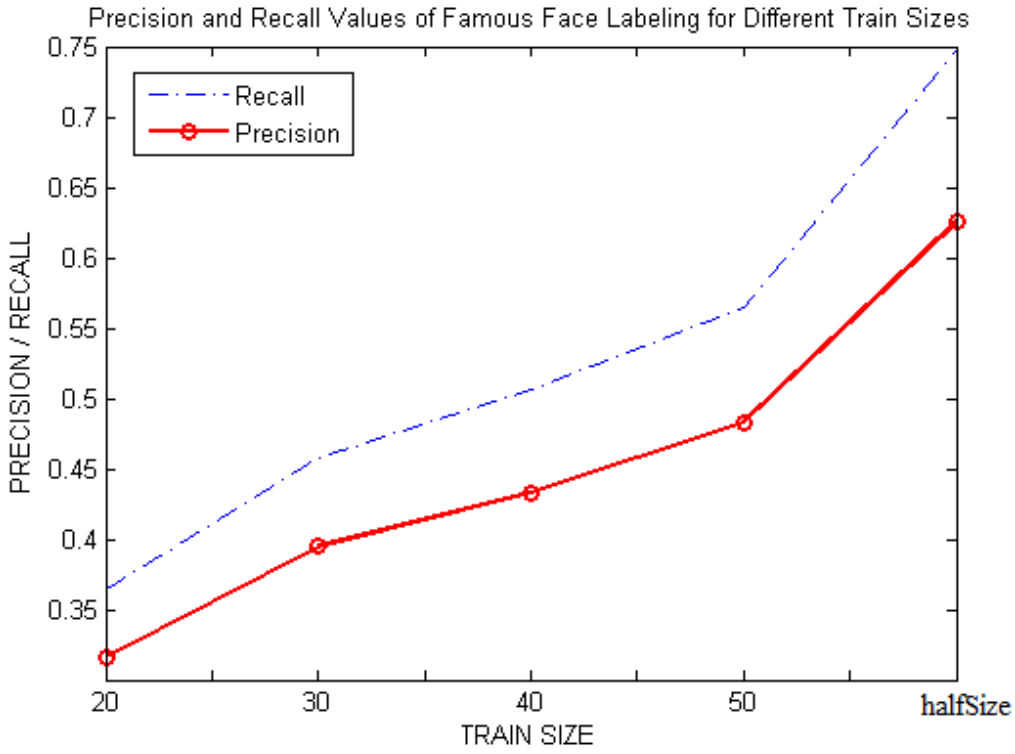

Figure 5.5: Precision and Recall Values for Popular Face Labeling vs TS.

And Figure [5.7] illustrates the average precision and recall values of outliers for different sizes of training data. In this figure, different than the previous one, the recall value decreases 3% as TS increases from 20, to half of the data size, *halfSize*. However; the precision value excessively increases as TS increases, and this is the result of the significant decrease in the number of false positive outliers (FP_Outliers). Figure [5.6] explores the decrease in FP_Outliers and FP_Famous. 17% of a decrease can be observed for the false positives of outliers. Our main contribution in this step, is to successfully determine the less-frequent faces among the face images, via correctly naming the most-frequently appearing ones. If the number of incorrect outlier labeling is low, we will have a higher accuracy rate for the next step of the algorithm.

These figures, illustrates that the accuracy rate of detection and labeling increases as the training size increases. Further in our experiments, rather than

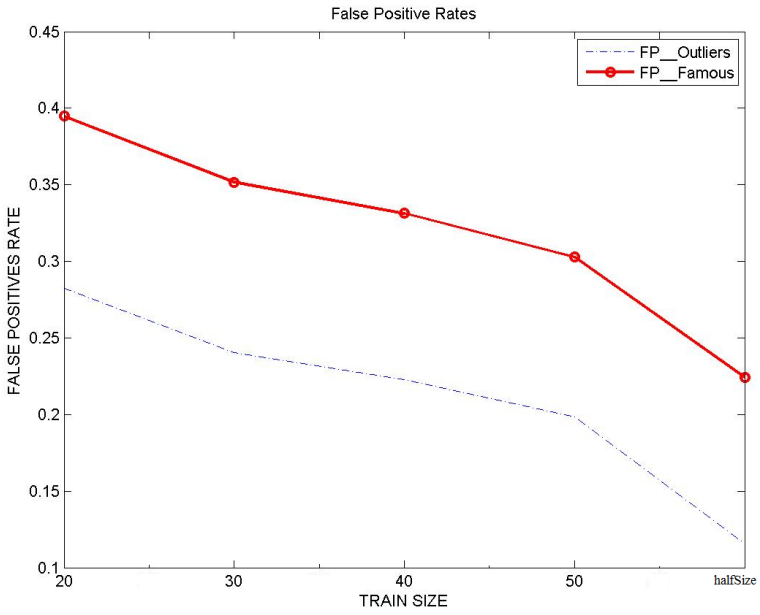

Figure 5.6: False Positive Rates of different TS values for Outlier Detection and Popular Face Labeling.

selecting TS to be 50, we prefer to take TS as *halfSize*, since there are categories of names with less than 50 samples. And; along with this reason, we get a higher accuracy rate for categories with more number of face samples.

Naming Less Frequently Appearing People on the Web

The experimental results for different values of TS in naming less-frequently appearing people on the Web, showed that; TS does not have a significant effect on the precision and recall values on this step. It rather affects the results indirectly, since the accuracy of naming less-known faces depends on the accuracy of the previous steps. As we match the outlier face images to less-frequently appearing names more accurately in previous steps, we will achieve more accurate labeling of less-frequently appearing faces in this step.

Figure [5.8] illustrates the precision and recall values for outlier detection and

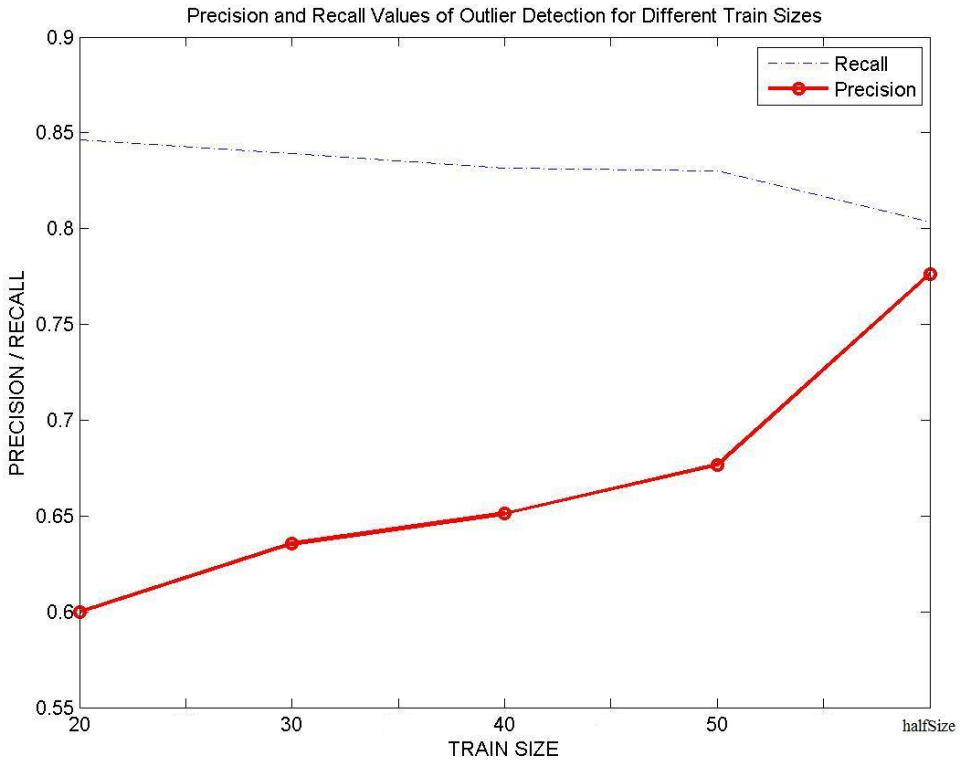

Figure 5.7: Precision and Recall Values for Outlier Detection.

labeling of less-known faces. Although we are not able to observe a consistent decrease or increase on the PR and REC values for different sizes of TS, we can make an evaluation based on Table [5.1]. Although there is an approximately 2% of a slight change in the recall values and around 6% of a change on the precision values, the reason for the decrease in recall values for less-known face labeling can be interpreted as follows. Judging by the results of table [5.1] the accuracy rate in detecting the outliers among the existing outliers, namely recall, decreases 4.26% as TS increases, on the other hand, since we used a supervised classification on popular face labeling, the recall value of popular face labeling increases to a great extent, 38.36%. The decrease in the outlier detection caused the number of face samples collected for a candidate less-known name to decrease (Figure[5.9]). This slight decrease on the samples collected, causes a decrease in accurately naming less frequently appearing faces.

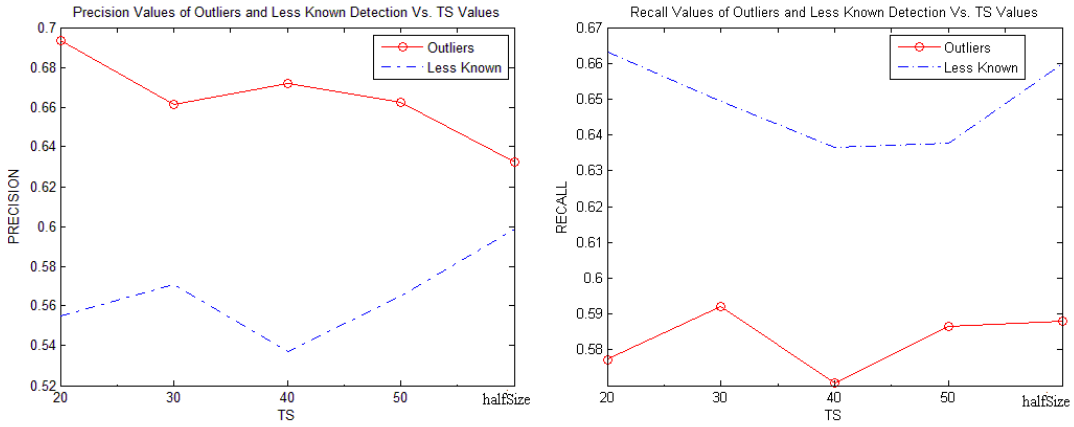

Figure 5.8: Precision and Recall Values for Outlier Detection and Labeling of the Less Frequently Appearing People vs TS.

5.4.1.2 Experimental results for different values of n

Following two subsections will give the experimental results on the steps of our algorithm for different values of *n*.

Naming Most Frequently Appearing People on the Web

The number of popular people names, and the number of less-known people names, are selected among 140 categories of names provided by PubFig dataset. As already explored in section 5.4 we used a 60% to 40% ratio between the number of popular and less-known name sets, respectively. We referred to the total number of popular and less-known face sets as *n*. For a better evaluation, we have chosen different sizes of *n*, n = { 50, 75, 100 } . Using the evaluation from the previous section, we selected TS to be *halfSize* for each category, since it gives a better accuracy rate.

Table [5.2] shows the precision and recall values of both outlier detection and popular face labeling for different sizes of n. (n= { 50, 75 and 100 }). As the number of categories for supervised classification increases, the correct labeling for

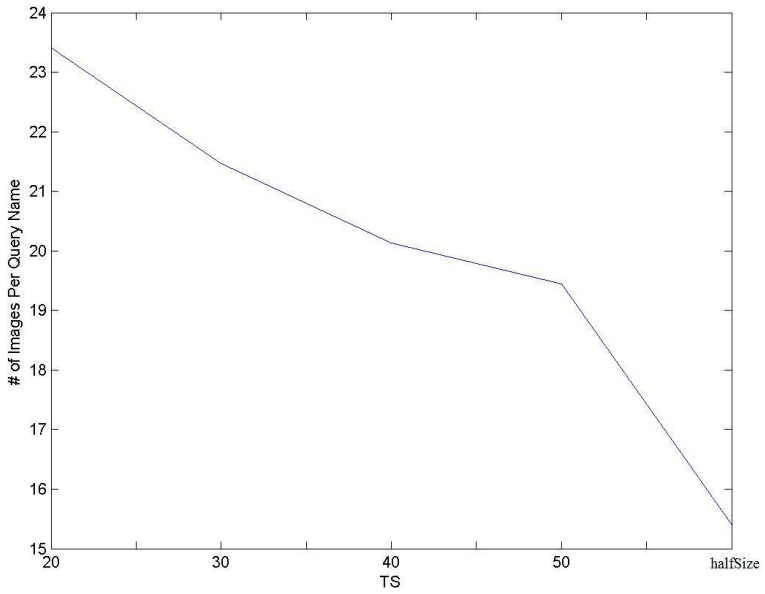

Figure 5.9: Number of Face Images Collected for a Less-Known Name vs. Training Size.

popular face labeling decreases. As illustrated on the Table 5.2, the precision and recall values for popular face labeling (PR_Famous and Rec_Famous) decrease, as n increases.

Table 5.2: Average Evaluation Results For Different n Values

n	PR_Outlier	PR_Famous	Rec_Outlier	Rec_Famous
50	77.6375	62.5487	80.36	74.8
75	77.143	62.5341	83.74	72.68
100	76.0217	61.3654	85.22	70.42

Naming Less Frequently Appearing People on the Web

In the previous section, the experimental results for less-known face detection are explored. Based on these experimental results, in this section, we will evaluate the results from the previous algorithm, which is to collect the detected outlier face

images and match them with corresponding less-known face names extracted from their textual contents. The outliers, in other words the less-known faces detected in the first step, are matched with their corresponding candidate less-known face name. For each less-known face name, bunch of face images are collected as a result of the first step. This collection of less-known face images may or may not be relevant with the name. In this step, the experimental results of pruning the irrelevant face images among this collection will be explored.

As indicated in Section 3.2.2, a similarity graph is constructed for the collected images, and an algorithm similar to Borda Rank Algorithm [1] given in Algorithm [1] is executed. The precision and recall values of both outlier detection, and labeling of the less-known names for n = { 50, 75, 100 } for 10-fold cross validation is illustrated in Figures [5.10] and [5.11].

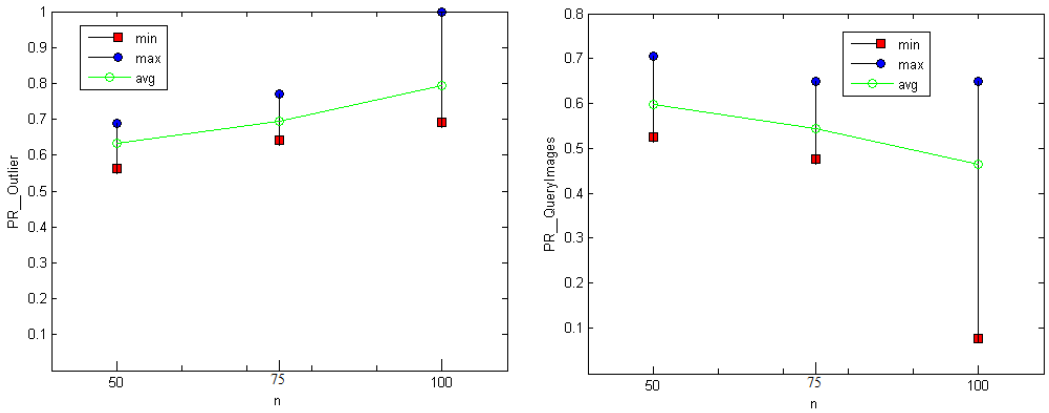

Figure 5.10: Precision Values for Outlier Detection and Less-Known Face Labeling.

We have calculated the average results for precision and recall values of 10-fold cross validation on the detection of irrelevant face images, or namely the outliers, and labeling of the less-known face images. Figure [5.12] and Figure[5.13] explores the average results for precision and recall values respectively for the face images detected as irrelevant(outlier) and relevant(query image).

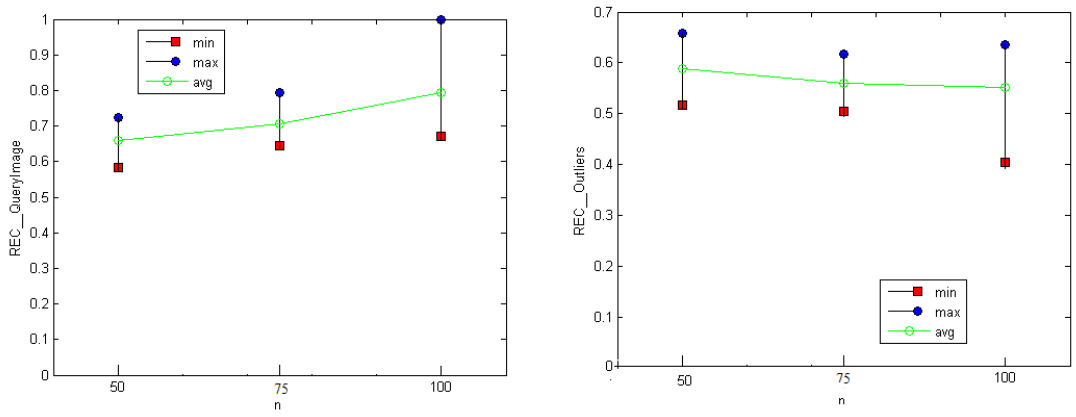

Figure 5.11: Recall Values for Outlier Detection and Less-Known Face Labeling.

In this step of the algorithm, our main contribution is to find the faces belonging to the less-known people names. So that, as a result we will be able to collect face images automatically for a less-known name. As it can be observed from Figure [5.13], the recall value for less-known face labeling increases as n increases, however, the precision value decreases to a great extent. Although the accuracy of labeling the less-known faces among the existing less-known face images (recall), increases, the accuracy of the correct labeling among the labeled less-known face images (precision) decreases. This means, the false positives for less-known face naming increases as the number of selected name sets, n, increases. Since a total of 500 images are generated for multi-face images, the number of images collected for a less-known name, decreases; as we increase the number of popular and less-known name category sets. Figure [5.14] shows the decrease in the average number of images collected for a less-known name as n increases. As a result of this decrease, accuracy in less-known face labeling is affected negatively.

5.4.1.3 Experimental results for different values of text generation probability

In this subsection, the experimental results for the variations in the probability value calculated for random text generation will be examined, by keeping the

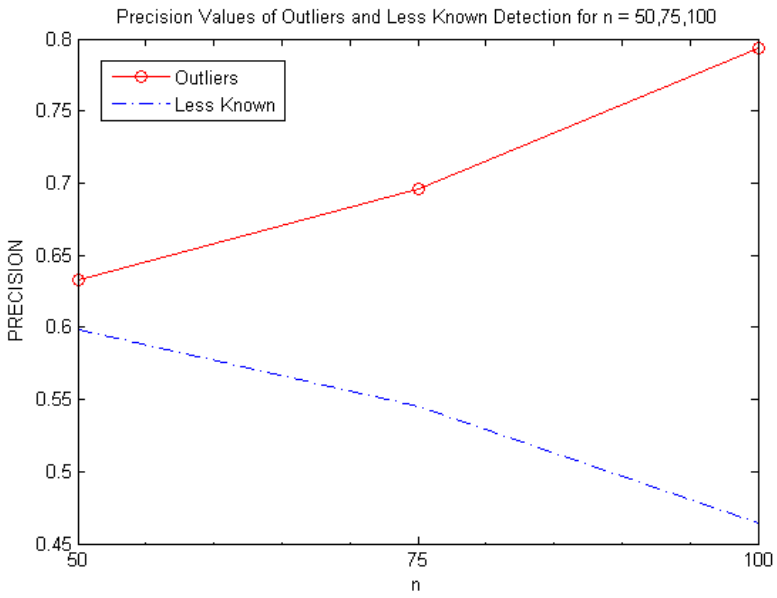

Figure 5.12: Average Precision For Outlier Detection and Less-Known Face Labeling vs. n.

variables n and TS stable, (n = 50, TS = *halfSize* for each category). The results in this experiment will affect the less-frequently name labeling since we are changing the variable for the text generation procedure. As explained in chapter 4, the texts are generated in the following way. For an image with s faces,$(s+1)$ names will be generated, where the first s names will be generated for s faces respectively, and the last slot will be a location or any other proper name if it exists with a 50% probability. The s names, representing the s faces, will randomly be correct or incorrect in terms of actually belonging to the face it represents. This random selection of being correct or incorrect is based on a probabilistic value. In our experiments, we have selected this correct/incorrect ratio respectively to be 80%-20%, 70%-30%, 60%-40%. With the intuition of a face appearing generally when his or her name is mentioned, we kept the probability of having a correct name higher than having an incorrect one, for a face. From this point on, we will refer to the probability of having a correct name for a face as *Prob_Corr*.

When the correct name generation for face images decreases, the number of

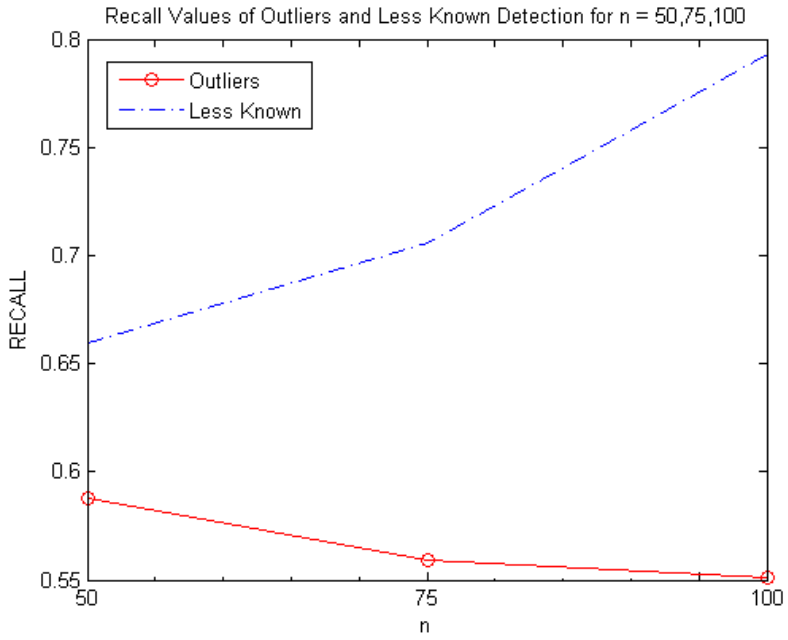

Figure 5.13: Average Recall For Outlier and Less-Known Faces vs. n.

incorrect face-name matching for less-frequently appearing people will increase. Therefore, the collection of less-known faces matched for each extracted name will contain less number of relevant face images. As a result of this situation, elimination of irrelevant faces among the less-known face collection for a name, will give a less successful result for lower Prob_Corr values. Figure 5.15 illustrates the increase in the rate of correct faces gathered for an extracted name for increasing probability values of correct name generation. Given this increase in the number of correct face collection for an extracted name, Figure 5.16 explores the increase in correct labeling of the existing correct face collection, in other words, the recall of the less-known face labeling and depending on this, the recall of outlier detection.

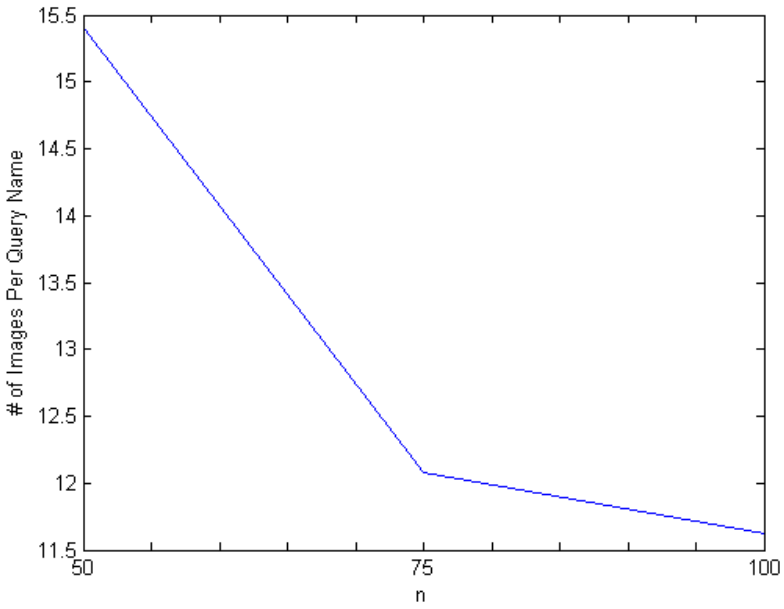

Figure 5.14: Number of Images Per Query for Different Values of n.

5.4.1.4 All Faces Taken as Outliers

In this step of our experiments, we compared the results where any face image is considered as an outlier. In other words, we evaluated the results where each face image, either being a famous face, or being a less known face, is labeled with the methodology used in the second step of our algorithm, which is used to name the less-frequently appearing faces. Then we compared the results of labeling the faces of famous people with both methods.

Table 5.3 compares the results of famous face labeling for two different methods. PR_Famous and Rec_Famous are the results of famous face labeling with classification method, whereas, PR_QueryImg_Famous and Rec_QueryImg_Famous are the results where famous faces are labeled via the method used for labeling less-known faces.

The results are evaluated for TS = *halfSize*, n = 50 and Prob_Corr = 80%. As expected, the labeling procedure for famous faces with classification method

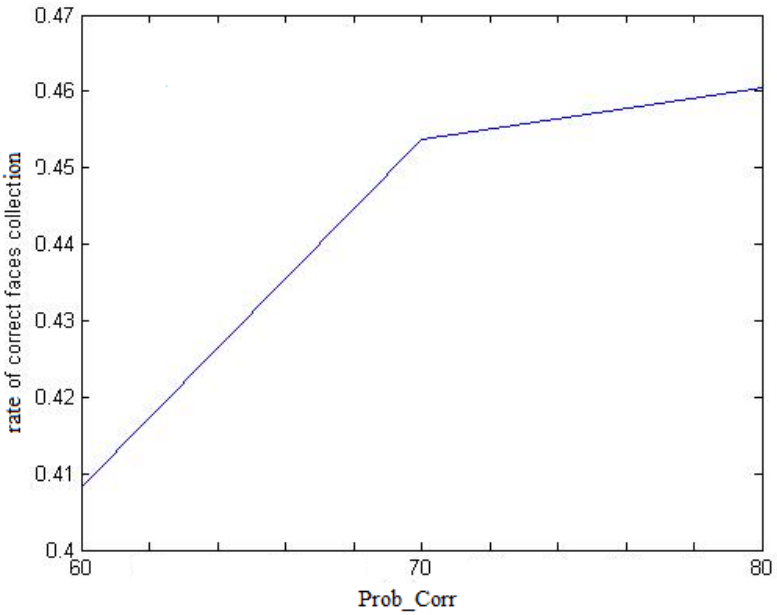

Figure 5.15: Rate of Correct Face Images Collected for a Less-Known Name vs. Prob_Corr.

Table 5.3: Average evaluation results for face labeling if all the faces are considered to be less-known faces.

PR_QueryImg_Famous	PR_Famous	Rec_QueryImg_Famous	Rec_Famous
51.90	62.54	67.58	74.8

outperforms the method used to label less-known people, in other words, the method where faces are considered to be less-known people, namely the outliers. On average, the classification method gives approximately 10% better accuracy results on labeling.

5.4.1.5 A specific result from selected values

In order to clarify the results of our algorithm, we will examine the following results which are taken from one of our experiments where 500 two-face images are generated for n=50, TS = *halfSize*. We will focus on the results from the

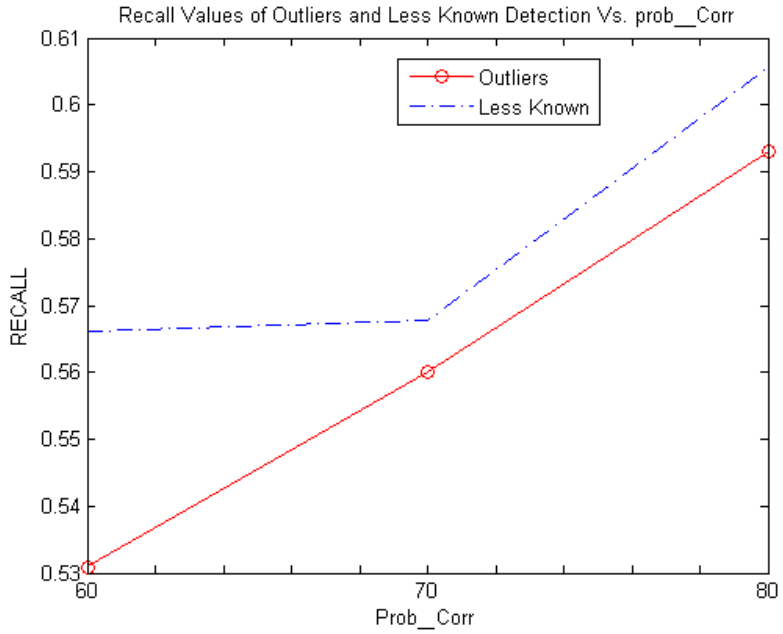

Figure 5.16: Recall Values for Outlier detection and Less-Known Name Labeling vs. Prob_Corr.

two-face images where the less-known person is Alec Baldwin (see Figure 5.17). Among the 500 randomly generated images, there are 24 images belonging to Alec Baldwin as being less-known person face. Among those 24 images, 21 of them are successfully labeled as less-known faces, or in other words as outliers, however 3 of them are labeled from the popular face name sets. 3 of the incorrectly labeled face images of Alec Baldwin are confused with Gene Hackman, Aaron Eckhart and Cristopher Walken, respectively.

For each image a random text is generated, either containing the name of the faces on the images or not, depending on the pre-determined probabilistic values. (in this case Prob_Corr = 0.8) Some of the generated names for corresponding images are illustrated in Figure 5.18.

For each detected outlier, names not matched with any face yet, are extracted from their textual content and matched with the detected outlier face images.

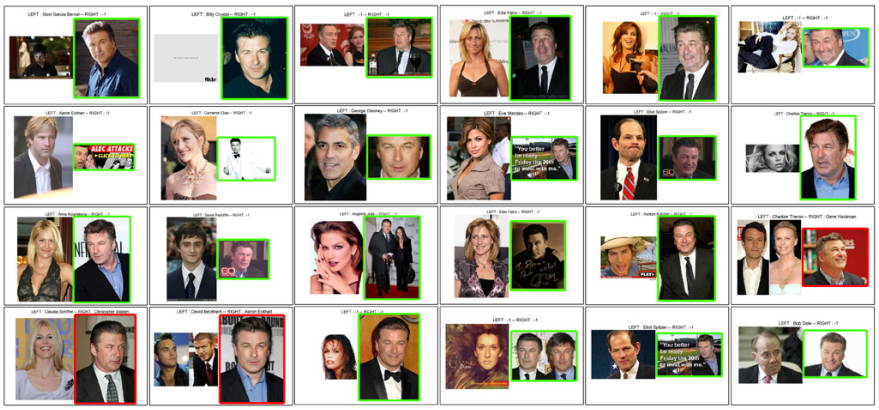

Figure 5.17: Images where Alec Baldwin is a less-frequently appearing people.

Figure 5.18: Generated names for corresponding images.

Figure 5.19 illustrates the face images collected for less-known name “Alec Baldwin”. During the process of the system for 500 images, using the textual content, 24 images matched to name “Alec Baldwin”, 17 of them actually belong to Alec Baldwin, while the remaining 7 images are irrelevantly matched to the name just because it appeared in their textual content.

Using our pruning algorithm for irrelevant images, the results are illustrated in Figure 5.19. Among the 7 irrelevant images, 4 of them are labeled as outlier, however 2 of the images belonging to Alec Baldwin also labeled as outliers. On the other hand, 17 images are labeled with the name Alec Baldwin; however, 15 of the images actually belong to him. Therefore, the results in terms of precision and recall for labeling the matched images with less-frequently appearing names are as follows. Precision of labeling a face in this collection with the name “Alec

Figure 5.19: Images matched for the name “Alec Baldwin”.

Baldwin” is 15/17 and the recall is also 15/17. (The precision and recall results are found to be equal by coincidence; since, the number of actual outliers in the collection and the number of faces labeled by the system are both 17.)

5.5 Using FW dataset

We first applied our algorithm on a subset of Faces in The Wild Dataset (FW) [3]. The dataset contains; the original news photos, their captions and the detected faces on these images. Each face in the dataset is labeled by the algorithm they proposed. 23 categories of names have been selected as a subset from FW for our algorithm. We first, use the labels they assigned to each face image as ground

truths, and then collect the face images labeled with the selected 23 categories. Since the labels were assigned by the algorithm proposed by [3], they were error-prone. Therefore, after collecting face images, a manual elimination is applied in order to eliminate the irrelevant faces from each category. Table 5.4 shows the number of face images left for 23 categories after the manual elimination.

Table 5.4: Number Of Images Per Query

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
53	58	50	67	71	54	45	63	54	94	88	97
13	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	
65	113	146	93	186	213	195	317	426	82	740	

Table 5.5: Query Names

1	Abdullah Gul		**13**	Hans Blix
2	Roh Moo-Hyun		**14**	Jean Chretien
3	Jiang Zemin		**15**	Hugo Chavez
4	David Beckham		**16**	John Ashcroft
5	Silvio Berlusconi		**17**	Ariel Sharon
6	Gray Davis		**18**	Gerhard Schroeder
7	Lula Da		**19**	Donald Rumsfeld
8	John Paul		**20**	Tony Blair
9	General Kofi		**21**	Colin Powell
10	Jacques Chirac		**22**	Saddam Hssein
11	Vladimir Putin		**23**	George Bush
12	Junichiro Koizumi			

In order to apply our algorithm, the original news images belonging to 23 people are selected. Among these images, we collected the ones with multi-face images, so that the faces belong to 23 people will be the faces of popular people, and the other faces on these images will belong to the less frequently appearing ones on the web. In the following sections experimental results for the steps of our algorithm will be explored.

5.5.1 Naming more frequently appearing people

In the web search engines, the results with the correct retrievals will be found at the very first pages. In this study, we have used a generated dataset. However; just using this dataset we may not always be able to find the sample images of a desired popular person. In these cases; one can use web image search engines and the retrievals of the first two pages, for consistency. With this intuition, we decided to select our training size as 20 at this step of the algorithm. For each category 20 random images are selected to be the training data, and the model for classification is trained with SVM algorithm.

In the FW dataset, there are multi-face images, their captions, and the detected faces on these images. After the elimination of multi-face images for 23 desired name, we make a second elimination on whether or not; the FW dataset contains the face images on this multi-face images. In other words, whether or not, their algorithm detected each face on this multi-face image subset. We have left with 63 multi-face images, a total of 130 face images, which will be the test set for our algorithm. Among these 130 face images, 73 of them belong to less-frequently appearing people, while 57 of the faces are from the selected 23 categories.

We have used two different ways in classification, one-class and, multi-class classification.

5.5.1.1 One class classification

In one class classification method of SVM, a binary classification is performed; therefore in our case we have labeled our training data, as *query* or *non-query* person. This approach is much like a one against all approach, where a name among 23 categories will be labeled as *query-person* while all the other face images will be labeled as *non-query*. To have a close number of samples from both categories (query and non-query), we have doubled the size of query-person images. 40 random images are selected from one category as a *non-query person*,

Figure 5.20: Labeled faces and detected outliers.

and 40 random samples from the rest of the 22 categories are selected as the samples of non-query person. Than the model is trained with one-class SVM classification, and the classification of 130 face images for the selected category is performed. This one-against-all procedure is performed for 130 test face-images from 23 popular face categories and outliers. For each 23 categories, 130 face-images are labeled as query or non-query element. The next step will be to merge the results of each 23-one-class classification.

For each category, i, the one-class SVM classification algorithm, gives an output of 130x1 matrix *predictedLabels*, which is a matrix having labels of *-1* and *1*, where *-1* means the image at index j is a non-query person, and *1* means, the image at index j belongs to category i . As a result, we will have a 130x23

matrix, *allPredictedLabels*, where each column i, represents the 130x1 binary column vector of label results for whether or not the images belong to category i. In order to merge the results we first check whether an image is labeled with more than one category. Among our experiments, we did not encounter such a problem; therefore, finally, we merged the 130x23 matrix, allPredictedLabels, into 130x1 matrix. For each image j, ($1 <= j <= 130$) we assigned the j^{th} index of the predictedLabels matrix, to index i in the j^{th} column of allPredictedLabels, if *allPredictedLabels(i,j)* is not *-1*. The indices, which are not labeled with any of the 23 categories, will remain as *-1*, in other words they will be assigned as the outliers, or the less frequently appearing people. (Algorithm [2])

Algorithm 2 Labeling using the output of one class svm

1: Input : allPredictedLabels : 130x23 matrix for output labels for binary classification of 23 categories
2: Output : predictedLabels the overall label results for total 130 images.
3: **for** each column i of $allPredictedLabels$ **do**
4: $rowPredicted$ = row i of $allPredictedLabels$ (1x23 matrix)
5: j = find the index of the value that is not equal to -1 in $rowPredicted$
6: $predictedLabels(i) = j$
7: **end for**

As a result, among 130 face images, containing 73 less-known face, and 57 popular face, 91 of the faces are labeled as outliers, and 39 of them are labeled from the popular 23 name categories. Among the 91 face images detected as outlier, 73 of them are actually outliers, namely the true positives for outliers, TR_Outlier, is 73. On the other hand, all the 39 face images labeled as popular face are correctly labeled, however, 18 of the popular face images are labeled as outliers. In other words, the number of false positives for outliers FP_Outlier, or the number of true negatives for popular face detection is 18. The precision and recall values for outlier detection and popular face labeling is given in table [5.8].

Consequently, 100% of the outliers are correctly found as outliers by the system, however; the system, labeled 18 of the popular faces as outliers as well. On the other hand, the system labeled 39 of the face images correctly, in other words, 39 of the 53 face images labeled correctly, while 18 of them are labeled as outliers. Although there seems to be a 100% of accuracy rate for precision value of outlier

Table 5.6: Precision and Recall Values of FW

no of Detected Outliers	91
no of Correctly Detected Outliers (TP_Outlier)	73
no of Outliers	73
PR_Outlier	100%
REC_Outlier	80.22%
no of Labeled Famous Faces	39
no of Correctly Labeled Popular Faces (TP_Famous)	39
no of Popular Faces	57
PR_Famous	68.42%
REC_Famous	100%

detection, and recall value of popular face labeling, the system incorrectly labeled 13.85% of the face images.

5.5.1.2 Multi-class classification

In multi-class classification, we have used multi-classification SVM provided by LibSVM. In this step, 20 random face images for each category are selected to be the training data for training the SVM model. Then the 130 face images are given as the input test data. The result of the algorithm returns a 130x1 matrix label along with the 130x23 probability estimate matrix having the probabilistic results for each 130 images to be in one of 23 categories. Each image is labeled with the index of the maximum probabilistic result. However; contrarily to the previous method, each image is labeled with one of the 23 categories. In other words, at the first step, there is no outlier detection. To handle this problem, we have used the 130x23 probability estimate matrix and a threshold value for deciding whether or not an image should be labeled with its highest probabilistic

response. First for each image, the highest probabilistic values are kept, and then we have used two different threshold values which will be explained below.

- **Threshold 1 (THRS1)**:

The 1x23 probability estimate matrix, for an image, keeps the values of probabilistic results for that image to belong any of 23 categories. In other words, the 130x23 probability estimate matrix, *PE*, keeps the probability of i^{th} image to be in the j^{th} category in *PE(i,j)*. For each image *i*, we keep the highest probability result. The mean of these results is taken to be the first threshold value for our outlier detection. The probabilistic results below this threshold is selected to be the outliers, and the probabilistic results greater than the threshold, are labeled with the category of its highest probabilistic response. However; using simply mean value as a threshold, does not give us the desired result successfully in labeling, and especially, detecting the outliers. The results for mean threshold value are given in Table [5.7]

Table 5.7: Precision and Recall Values of FW

no of Detected Outliers	90
no of Correctly Detected Outliers (TP_Outlier)	60
no of Outliers	73
PR_Outlier	67.50%
REC_Outlier	47.37%
no of Labeled Popular Faces	40
no of Correctly Labeled Popular Faces (TP_Famous)	27
no of Popular Faces	57
PR_Famous	66.67%
REC_Famous	82.19%

- **Threshold 2 (THRS2)**:

Our second threshold value is selected to be a value above the mean value calculated for the first threshold. As a second approach, we have selected the highest probabilistic estimates, above the threshold, and get the mean of these values. The second threshold value is the average of the two mean values calculated. (Algorithm [3])

Algorithm 3 Labeling using the output of one class svm

1: PE the 130x23 probability estimate matrix
2: $maxPEVals$ = get the maximum values in PE // result is a 130x1 matrix
3: $meanMaxPE$ = get the mean value of $maxPEVals$
4: $valuesAboveMean$ = get the values above mean in $maxPEVals$
5: $meanMaxPE2$ = get the mean value of $valuesAboveMean$
6: $threshold2 = (meanMaxPE + meanMaxPE2) / 2$

This threshold value gives better results in labeling and especially in outlier detection.

Table 5.8: Precision and Recall Values of FW

no of Detected Outliers	76
no of Correctly Detected Outliers (TP_Outlier)	55
no of Outliers	73
PR_Outlier	66.67%
REC_Outlier	63.16%
no of Labeled Popular Faces	54
no of Correctly Labeled Popular Faces (TP_Famous)	36
no of Popular Faces	57
PR_Famous	72.37%
REC	75.34%

When the two threshold values are compared, the second threshold value gives better results; since THRS1 < THRS2, using THRS2, we are able to prune more

number of outliers, as a result, the recall value of outlier, REC_Outlier gives better results for second threshold value.

5.5.2 Naming less-frequently appearing people

For the final step of the algorithm, the candidate names for less-known people are extracted from textual content. However; FW dataset were not adequate for multiple appearances of the same less-known people, therefore, we were not be able to collect bunch of face images for the names extracted from textual content.

As a result, we were not able to complete the last step of our algorithm, which is to find the most similar subset of face images collected under a less-known person name.

5.6 Comparison of the results from different facial features

As a final step, we would like to focus on the reason for selecting PubFig dataset and their facial features by comparing their classification results with the classification results on the facial features of FW dataset. In order to compare the two results and feature selection methods, we needed to perform the classification algorithm for both methods on the same dataset. We were not able to extract the PubFig Facial features for FW dataset, however, it was possible to extract the SIFT facial features on PubFig dataset. Therefore; we used PubFig Dataset for comparison, since it both provides the facial features and the detected faces on images. Using the detected faces on these images, the SIFT descriptors are extracted.

For 140 categories of names, the PubFig facial features for each face image are provided; and the detected faces on the images are given, as well. However; since PubFig only provides the URLs for their face image dataset, we were not able to

download some of the images that are not on the web anymore. The number of images collected for 140 categories is given in Table A.1 and A.2.

For each image the SIFT descriptors of the specific 9 facial features are extracted on the detected faces, and we already have PubFig facial features for corresponding face images. For each category, multi-class SVM classification is performed. The training set size is decided empirically. We perform SVM classification for training sizes of 10, 20, 30,.., and 60, and the rest of the dataset is used for testing. Figure [5.21] illustrates the accuracy rate for selected training size samples.

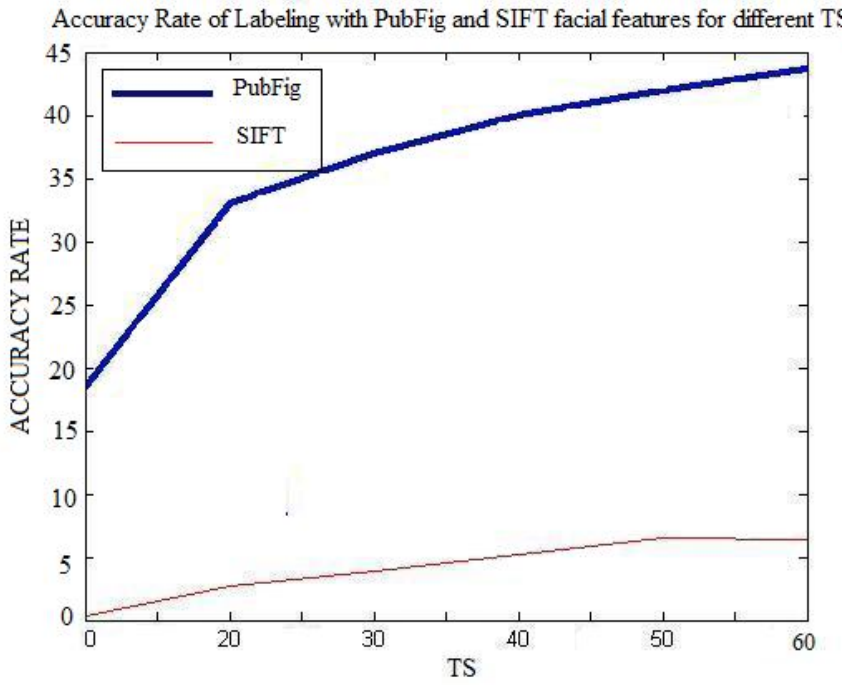

Figure 5.21: Accuracy Rate of Labeling for Attribute Classifiers and SIFT Facial Features.

Consequently, although in the previous section, the accuracy rates were higher when SIFT facial features used, when the number of categories for classification increase, the classification with SIFT facial features shows a low accuracy rate. As it is illustrated in Figure [5.21] PubFig Facial Features, outperforms the accuracy rate provided by Sift Facial Features on SVM classification to a great extent.

While the LibSVM accuracy results for SIFT facial features on 140 images change between 0.3% to 6.4% , the PubFig facial features on the other hand, increases from 18.43% to 43.68% as TS increases. Therefore; in order to get higher accuracy rate for labeling faces on images, we decided to use the PubFig facial features for face representations.

Chapter 6

Discussion

In this chapter, the discussion of our algorithm on different variables selected for execution will be introduced. During our experiments, we have executed the algorithm for different values of three variables, the training size for the SVM classification algorithm, TS; the total number of name sets selected both for popular and less-known names, n; and finally the probability value for either or nor generating a correct name for a face image in its textual content, Prob_Corr. Finally, we will discuss on the two different feature vectors we have used to represent faces.

6.1 Different values of training size (TS)

The accuracy rate of the supervised classification algorithms depends on the quality and size of the training data. The accuracy increases if the model is successfully trained; therefore higher training sizes give better results as expected. In our algorithm, we have used 6 different training sizes, TS = 10, 20, 30, 40, 50, the half size of the data for each category. Figure 6.1 illustrates the rate of change in accuracy for PubFig and FW datasets, as the training size increases.

As it can be observed from this figure, the accuracy rate increases, as the TS

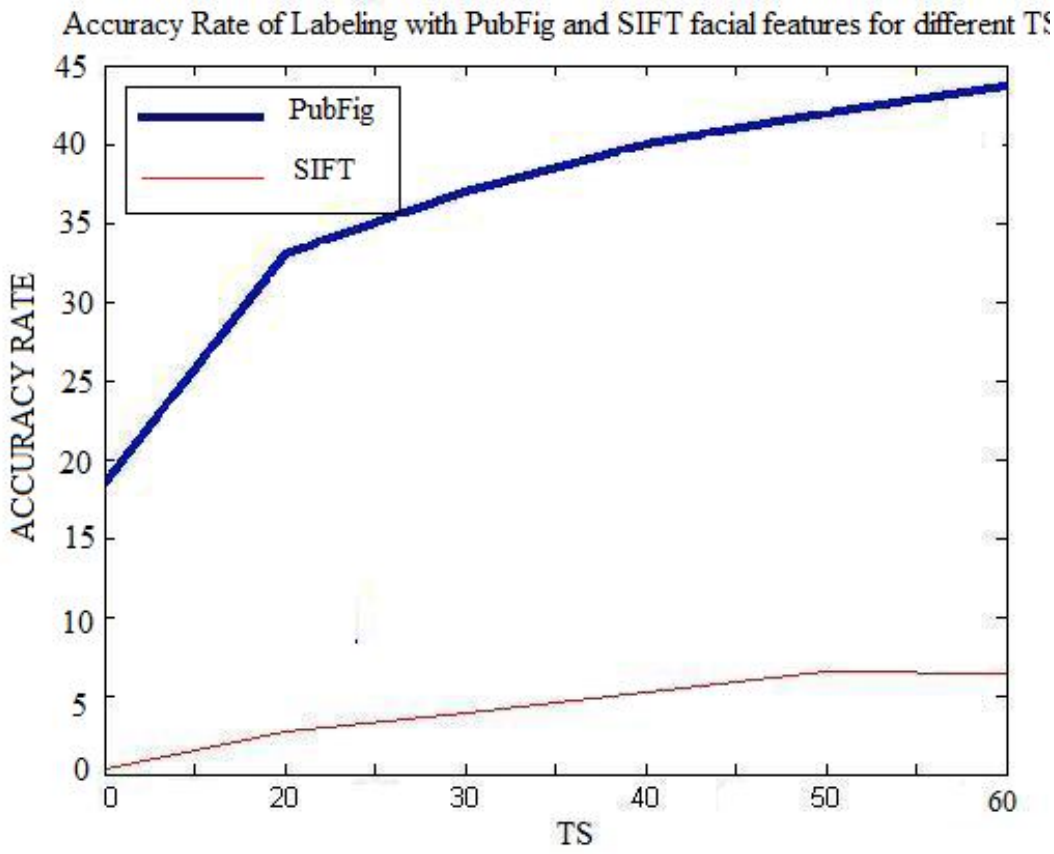

Figure 6.1: Accuracy rates for Pubfig and Sift face features vs. TS.

increases; however, after a point, the rate of change does not increase to a great extent. The last TS size gives a higher result compared to the first 5 training sizes. The reason is, for the last TS size we give half of the data for training; therefore, it's accuracy rate is excessively higher than the other results. However, when the first 5 TS are examined, the accuracy rate of change does not show a huge change after a point. Since our main contribution is to name the less-frequent faces, we wanted the first step of our algorithm which is to name more-frequently appearing faces, to give higher accuracy rates, as a result later in our experiments, TS is selected to be the half size of the data for each category.

6.2 Different values of name set size (n)

For evaluating our system, we have selected a total of n name sets, where 60% of this set belong to the popular faces, while the remaining 40% belong to the less-known faces. We have selected 3 different values for n, $n = 50, 75, 100$. The change in the number of name sets, mostly affected the results for labeling the less frequently appearing faces. Since we have selected randomly 500 images from both name sets to form a multi-face image, increasing the number of categories for less-frequent faces, results in having less number of samples for an extracted face name. However, since classification algorithms decrease in performance when the number of categories increases, a slight decrease in the success rate of labeling the more-frequent faces, occurs as well, when n changes. Therefore, for $n = 50$, 75, 100, size of the name sets for more frequently appearing faces will be 20,30, and 40. Figure (6.2) illustrates the accuracy, in other words the recall rate of popular face labeling. As the number of categories increases from 50 to 100, the recall value decreased 4.38%.

On the other hand, increasing n for the second step will result in having less number of samples for name categories of less frequently appearing faces. Figure 6.3 shows the number of images collected for an extracted name as n increases.

6.3 Different values of probability value for correct name generation (Prob_Corr)

For each multi-face image, we generated a textual content containing the names of the corresponding faces. However; in order to generate realistic textual contents, for each face image we decided to give a probability on whether or not the name generated will actually belong to that face. We have selected different probabilistic values for a name being correctly generated for its corresponding face. The selected probabilistic values are prob_Corr = 60, 70, 80.

Decreasing the correct name generation for textual contents, the accuracy

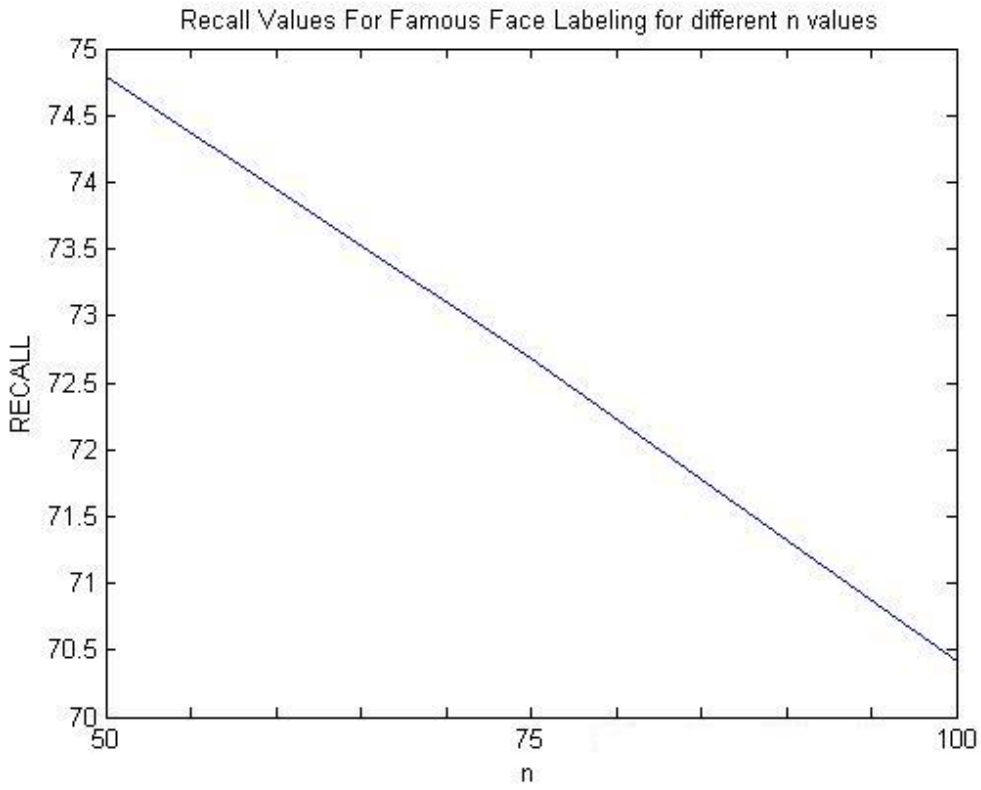

Figure 6.2: Precision and recall values for outlier detection.

rates of less-frequently appearing face labeling decreases as well. For each extracted name, less number of correct faces will be matched. As a result, the algorithm will not be able to correctly find the more similar face image set in the less-frequently appearing face collection. Figure 6.4 illustrates the increase in accuracy for increasing probability values.

6.4 Feature selection for face representation

We have executed our algorithm for two different facial feature methods. One is the commonly used SIFT descriptors extracted for 9 specific points on face, and the other is a novel approach introduced by Kumar et al. [15], PubFig

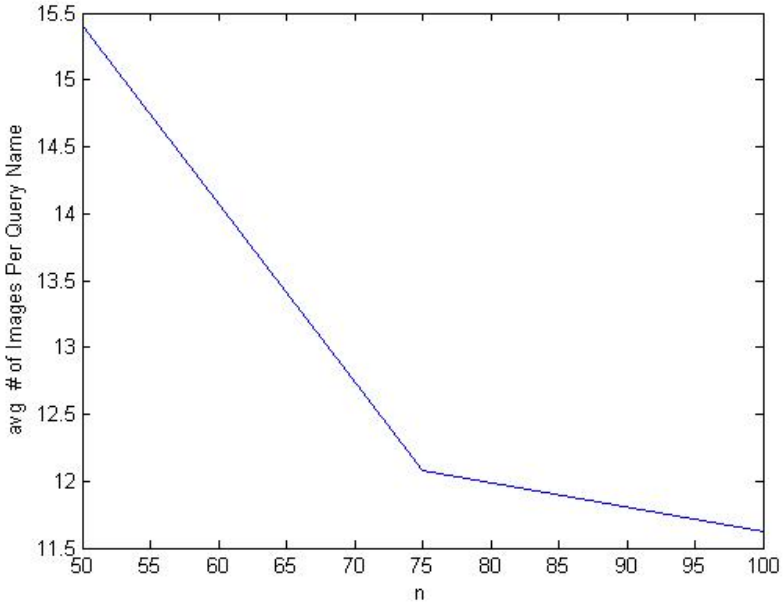

Figure 6.3: Number of images per query for different values of n.

Features, attribute and simile classifiers. As a result of our experiments, PubFig Face Representation outperformed the SIFT descriptors of 9 facial points. For the same set of images, accuracy rate of PubFig face representation on average is 31.48% better than the SIFT descriptor results.

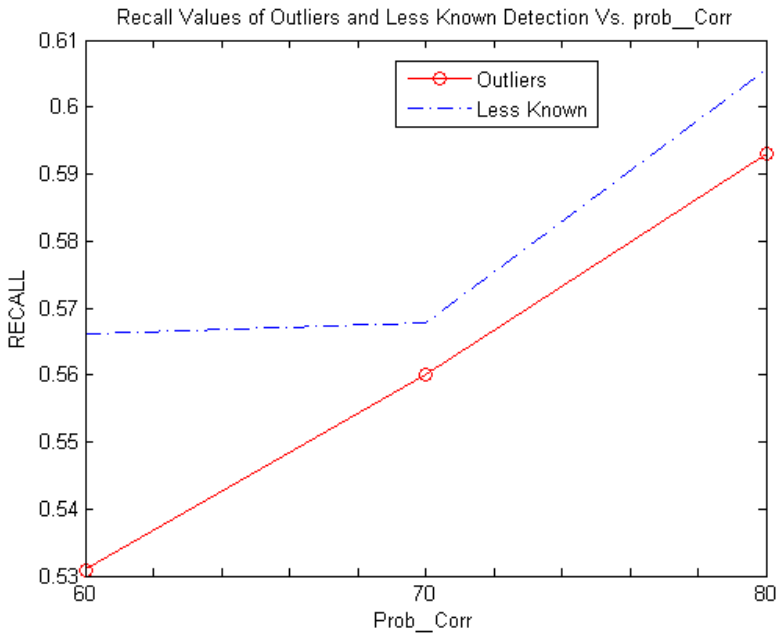

Figure 6.4: Accuracy rate for less-known face labeling and outlier detection vs. Prob_Corr.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we propose a method to name the less-frequently appearing people on the web, using both textual and visual contents, first by naming the more-frequent people on a multi-face image. In order to name the more frequent faces, SVM classification method is applied as a supervised classification method. Two different face representations are used for face images, one is the SIFT descriptors extracted for 9 specific points on a face, and the other is a novel method proposed by Kumar et al., the attribute classifiers explained in their study [15]. Using the results from SVM classification, more frequently appearing faces are named, and the less-frequently appearing faces are labeled as outliers. The textual contents of images are used to find the candidate labels for the faces labeled as outliers. From the textual contents, names that are not yet matched with more-frequently appearing faces, are extracted. Each outlier face image is labeled with those extracted names from the textual contents. As a result, for each extracted name, bunch of relevant and irrelevant images are collected. Since a person's name is mentioned around its face image, the majority of the collected images belong to their matching names of less-frequently appearing people. The irrelevant images collected for an extracted name are then, eliminated with an algorithm similar

to Borda Rank [1]. Finally, the face images that are not eliminated are labeled with the extracted name.

The experiments are performed on two different datasets. One is the FW dataset, the other is the PubFig dataset, each containing face images collected from realistic environments. FW dataset contains multi-face images and their textual content, however the PubFig dataset, solely contains the single face images. We first apply our algorithm on FW multi-face images, the more-frequent faces are labeled with an average precision of 66.67% and an average recall of 63.16%, on the other hand, the less-frequent faces are labeled as outliers with an average precision of 72.37% and an average recall of 75.34%; however, the less frequent faces collected for extracted names, were not adequate in amount to apply our algorithm for labeling them with their corresponding names. Therefore, to accomplish our studies we used PubFig Dataset.

PubFig Dataset, on the other hand, does not contain multi face images and their corresponding textual content. Therefore; using single face images, we randomly generate two-face images, one being a more-frequently appearing face, and the other being a less-frequently appearing face. Hence, a group of names are selected to be more-frequently appearing faces, while another group of names is formed to be the less-frequently appearing ones. The more-frequent faces are labeled with an average precision of 62.15% and an average recall of 72.63%, and the less-frequent faces are labeled as outliers with an average precision of 76.93% and an average recall of 83.11%. Among the less-frequent faces collected for an extracted name, the face images are labeled correctly with its corresponding name with an average precision of 59.88% and an average recall of 65.98%.

7.2 Future work

In this study, we have mostly focused on the process of labeling the less-frequent faces. However; improving the first step, which is to name the more-frequent faces,

will result in a better success rate for the second step, since the collected less-frequent faces will be less error-prone. For the first step, multi-class probabilistic SVM classification is used. The SVM classification method, assigns probabilistic values for each input image to be one of the categories. The highest probabilistic response among those categories for the input sample is selected to be the label of that input sample. In order to detect the less-frequent faces, a threshold value is calculated. The input samples with maximum probabilistic response of categories below this threshold are not labeled with this category name, but rather labeled as outliers. Although, the results are pretty satisfying, the more-frequent name labeling can be altered as follows. Rather than directly assigning an input sample to a label of the category with highest probabilistic response above the outlier threshold, this probabilistic result will be kept to decide on whether or not the face image should be assigned to the label of the highest response. Since the textual content of an image is provided, each label assigned by the SVM classifier will be searched in this textual content. If the label assigned by the SVM classifier exist in the textual content as well, a new higher probabilistic value will be assigned for its probabilistic result rather than the value assigned by SVM. With this approach, more-frequently face labeling will be improved.

For the second step of our algorithm, in order to find the relevant subset among the collected face images of a less-known person name, an algorithm similar to Borda Rank algorithm is applied on the similarity graph of these face images. With this algorithm, each face image is given a rank according to its similarity to the rest of the collection. The irrelevant images, namely the outliers, are eliminated via a threshold value calculated. The threshold value is calculated using simple formulas depending on the mean values. Several threshold values are calculated, and as a result, depending on the accuracy rates we decided on one method. Rather than, using values empirically selected among simple threshold formulas, graph based methods for finding the densest subset in a collection of images will be used in order to construct a more automatic system that would give higher accuracy rates.

Bibliography

[1] N. Ailon. Aggregation of partial rankings, p-ratings and top-m lists. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '07)*, pages 415–424, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[2] Alias-i Inc. LingPipe, http://www.alias-i.com/lingpipe/.

[3] T. Berg, A. Berg, J. Edwards, M. Maire, R. White, Y. Teh, E. Learned-Miller, and D. Forsyth. Names and Faces. *University of California Berkeley. Technical report*, 2007.

[4] M. Bicego, A. Lagorio, E. Grosso, and M. Tistarelli. On the use of sift features for face authentication. In *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, page 35, Washington, DC, USA, 2006. IEEE Computer Society.

[5] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[6] P. Duygulu, K. Barnard, N. de Freitas, P. Duygulu, K. Barnard, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary, 2002.

[7] M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... buffy automatic naming of characters in tv video. In *BMVC*, 2006.

[8] M. Everingham, J. Sivic, and A. Zisserman. Taking the bite out of automatic naming of characters in TV video. *Image and Vision Computing*, 27(5), 2009.

[9] J. H. Friedman. Another approach to polychotomous classification.

[10] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Automatic face naming with caption-based supervision. pages 1–8, 2008.

[11] M. E. Houle. *NII Technical Report (NII-2006-008E)*, 2006.

[12] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[13] N. Ikizler and P. Duygulu. Person search made easy. pages 578–588, 2005.

[14] U. H.-G. Kre. Pairwise classification and support vector machines. pages 255–268, 1999.

[15] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and Simile Classifiers for Face Verification. In *IEEE International Conference on Computer Vision (ICCV)*, Oct 2009.

[16] D.-D. Le and S. Satoh. Unsupervised face annotation by mining the web. In *ICDM*, pages 383–392, 2008.

[17] D.-D. Le, S. Satoh, M. E. Houle, and D. P. T. Nguyen. Finding important people in large news video databases using multimodal and clustering analysis. In *ICDEW '07: Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop*, pages 127–136, Washington, DC, USA, 2007. IEEE Computer Society.

[18] C. Liu, S. Jiang, and Q. Huang. Naming faces in broadcast news video by image google. In *MM '08: Proceeding of the 16th ACM international conference on Multimedia*, pages 717–720, New York, NY, USA, 2008. ACM.

[19] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91, Nov. 2004.

[20] T. Miller, A. C. Berg, J. Edwards, M. Maire, R. White, Y.-W. Teh, E. Learned-Miller, and D. Forsyth. Faces and names in the news. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[21] D. Ozkan and P. Duygulu. A graph based approach for naming faces in news photos. pages II: 1477–1482, 2006.

[22] P. T. Pham, M.-F. Moens, and T. Tuytelaars. Cross-media alignment of names and faces. *IEEE Transactions on Multimedia*, 12(1):13–27, Jan. 2010.

[23] S. Satoh and T. Kanade. Name-it: Association of face and name in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pages 368–373, 1997.

[24] J. Sivic, M. Everingham, and A. Zisserman. "Who are you?" - Learning person specific classifiers from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1145–1152, 2009.

[25] Y. Su, S. Shan, X. Chen, and W. Gao. Hierarchical ensemble of global and local classifiers for face recognition. *Trans. Img. Proc.*, 18(8):1885–1896, 2009.

[26] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Real-Life Images workshop at the European Conference on Computer Vision (ECCV)*, October 2008.

[27] J. Yang, M.-Y. Chen, and A. Hauptmann. Finding person x: Correlating names with visual appearances. In *International Conference on Image and Video Retrieval (CIVR'04)*, Dublin City University, Ireland, July 21-23 2004.

[28] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Comput. Surv.*, 35(4):399–458, 2003.

Appendix A

PubFig Dataset

A.1 Evaluation Name Categories

140 PubFig Dataset evaluation name categories are given in Table A.1 and A.2.

A.2 PubFig Attribute Classifiers

65 selected attribute classifiers for facial features introduced for PubFig Dataset is illustrated in Table A.3 and A.4.

Table A.1: PubFig Name Categories and Number of Images Per Category

	Name	**Image Count**		**Name**	**image Count**
1	Aaron Eckhart	287	**36**	Dave Chappelle	63
2	Adam Sandler	309	**37**	David Beckham	374
3	Adriana Lima	277	**38**	Denzel Washington	229
4	Alberto Gonzales	228	**39**	Donald Trump	248
5	Alec Baldwin	283	**40**	Drew Barrymore	450
6	Alicia Keys	400	**41**	Dustin Hoffman	217
7	Angela Merkel	169	**42**	Edie Falco	111
8	Angelina Jolie	1091	**43**	Eliot Spitzer	233
9	Anna Kournikova	321	**44**	Eliza Dushku	258
10	Antonio Banderas	206	**45**	Eva Mendes	471
11	Ashley Judd	228	**46**	Gael Garcia Bernal	268
12	Ashton Kutcher	311	**47**	Gene Hackman	92
13	Avril Lavigne	792	**48**	George Clooney	542
14	Ben Affleck	311	**49**	George W Bush	162
15	Beyonce Knowles	448	**50**	Gillian Anderson	267
16	Bill Clinton	144	**51**	Gisele Bundchen	194
17	Billy Crystal	115	**52**	Gordon Brown	148
18	Bob Dole	66	**53**	Gwyneth Paltrow	522
19	Brad Pitt	1086	**54**	Halle Berry	490
20	Brendan Fraser	217	**55**	Harrison Ford	343
21	Bruce Willis	274	**56**	Holly Hunter	72
22	Cameron Diaz	489	**57**	Hugh Grant	340
23	Carla Gugino	157	**58**	Jack Nicholson	207
24	Carson Daly	85	**59**	James Franco	301
25	Cate Blanchett	415	**60**	James Gandolfini	193
26	Celine Dion	236	**61**	Jason Statham	187
27	Charlize Theron	529	**62**	Javier Bardem	159
28	Chris Martin	76	**63**	Jay Leno	195
29	Christopher Walken	127	**64**	Jeff Bridges	73
30	Cindy Crawford	178	**65**	Jennifer Aniston	617
31	Claudia Schiffer	181	**66**	Jennifer Lopez	484
32	Colin Farrell	422	**67**	Jennifer Love Hewitt	372
33	Colin Powell	352	**68**	Jeri Ryan	153
34	Daisy Fuentes	79	**69**	Jerry Seinfeld	179
35	Daniel Radcliffe	881	**70**	Jessica Alba	571

Table A.2: PubFig Name Categories and Number of Images Per Category

	Name	**Image Count**		**Name**	**image Count**
71	Jessica Simpson	545	**106**	Noah Wyle	106
72	Jimmy Carter	161	**107**	Oprah Winfrey	449
73	Joaquin Phoenix	263	**108**	Orlando Bloom	1266
74	Jodie Foster	367	**109**	Owen Wilson	177
75	John Lennon	153	**110**	Philip Seymour Hoffman	119
76	John Malkovich	118	**111**	Quincy Jones	125
77	John Travolta	372	**112**	Ralph Nader	186
78	Jon Stewart	171	**113**	Ray Romano	69
79	Kate Moss	301	**114**	Reese Witherspoon	365
80	Kate Winslet	345	**115**	Renee Zellweger	344
81	Katie Couric	229	**116**	Ricky Martin	329
82	Keanu Reeves	309	**117**	Robert Downey Jr	130
83	Keira Knightley	566	**118**	Rod Stewart	118
84	Lance Armstrong	186	**119**	Rosario Dawson	328
85	Leonardo DiCaprio	659	**120**	Rosie Perez	85
86	Liam Neeson	177	**121**	Russell Crowe	297
87	Lindsay Lohan	1536	**122**	Salma Hayek	485
88	Liv Tyler	298	**123**	Shania Twain	234
89	Lucy Liu	306	**124**	Sharon Stone	425
90	Mariah Carey	332	**125**	Shinzo Abe	71
91	Martha Stewart	247	**126**	Sigourney Weaver	205
92	Matt Damon	398	**127**	Silvio Berlusconi	238
93	Matthew Broderick	151	**128**	Simon Cowell	366
94	Mel Gibson	393	**129**	Steven Spielberg	237
95	Meryl Streep	353	**130**	Susan Sarandon	231
96	Michael Bloomberg	250	**131**	Tiger Woods	170
97	Michael Douglas	193	**132**	Tina Fey	270
98	Mikhail Gorbachev	131	**133**	Tom Cruise	519
99	Minnie Driver	151	**134**	Tom Hanks	281
100	Monica Bellucci	251	**135**	Tony Blair	173
101	Morgan Freeman	311	**136**	Tyra Banks	378
102	Nathan Lane	69	**137**	Uma Thurman	359
103	Nicolas Cage	320	**138**	Victoria Beckham	412
104	Nicolas Sarkozy	126	**139**	William Macy	100
105	Nicole Kidman	614	**140**	Will Smith	325

Table A.3: 65 Attribute Classifiers

1	Asian	**14**	Nose-Mouth Lines	**27**	Color Photo
2	Mouth Wide Open	**15**	Black Hair	**28**	Round Face
3	Attractive Woman	**16**	Obstructed Forehead	**29**	Curly Hair
4	Mustache	**17**	Blond Hair	**30**	Round Jaw
5	Baby	**18**	Oval Face	**31**	Double Chin
6	No Beard	**19**	Blurry	**32**	Semi-Obscured Forehead
7	Bags Under Eyes	**20**	Pale Skin	**33**	Environment
8	No Eyewear	**21**	Brown Hair	**34**	Senior
9	Bald	**22**	Posed Photo	**35**	Eye Width
10	Nose Shape	**23**	Child	**36**	Shiny Skin
11	Bangs	**24**	Receding Hairline	**37**	Eyebrow Shape
12	Nose Size	**25**	Chubby	**38**	Sideburns
13	Black	**26**	Rosy Cheeks	**39**	Eyebrow Thickness

Table A.4: 65 Attribute Classifiers

40	Smiling	**53**	Harsh Lighting
41	Eyeglasses	**54**	Visible Forehead
42	Soft Lighting	**55**	High Cheekbones
43	Eyes Open	**56**	Wavy Hair
44	Square Face	**57**	Indian
45	Flash Lighting	**58**	Wearing Hat
46	Straight Hair	**59**	Male
47	Frowning	**60**	Wearing Lipstick
48	Sunglasses	**61**	Middle-Aged
49	Goatee	**62**	White
50	Teeth Not Visible	**63**	Mouth Closed
51	Gray Hair	**64**	Youth
52	Teeth Visible	**65**	Mouth Partially Open