Distortion Robust Biometric Recognition

by

Jinane Mounsef

A Dissertation Presented in Partial Fulfillment
of the Requirement for the Degree
Doctor of Philosophy

Approved September 2018 by the
Graduate Supervisory Committee:

Lina Karam, Chair
Baoxin Li
Antonia Papandreou-Suppapola
Fengbo Ren

ARIZONA STATE UNIVERSITY

December 2018

ABSTRACT

Information forensics and security have come a long way in just a few years thanks to the recent advances in biometric recognition. The main challenge remains a proper design of a biometric modality that can be resilient to unconstrained conditions, such as quality distortions. This work presents a solution to face and ear recognition under unconstrained visual variations, with a main focus on recognition in the presence of blur, occlusion and additive noise distortions.

First, the dissertation addresses the problem of scene variations in the presence of blur, occlusion and additive noise distortions resulting from capture, processing and transmission. Despite their excellent performance, deep methods are susceptible to visual distortions, which significantly reduce their performance. Sparse representations, on the other hand, have shown huge potential capabilities in handling problems, such as occlusion and corruption. In this work, an augmented SRC (ASRC) framework is presented to improve the performance of the Spare Representation Classifier (SRC) in the presence of blur, additive noise and block occlusion, while preserving its robustness to scene dependent variations. Different feature types are considered in the performance evaluation including image raw pixels, HoG and deep learning VGG-Face. The proposed ASRC framework is shown to outperform the conventional SRC in terms of recognition accuracy, in addition to other existing sparse-based methods and blur invariant methods at medium to high levels of distortion, when particularly used with discriminative features. In order to assess the quality of features in improving both the sparsity of the representation and the classification accuracy, a feature sparse coding and classification index (FSCCI) is proposed and used for feature ranking and selection within both the SRC and ASRC frameworks.

The second part of the dissertation presents a method for unconstrained ear recognition using deep learning features. The unconstrained ear recognition is performed

using transfer learning with deep neural networks (DNNs) as a feature extractor followed by a shallow classifier. Data augmentation is used to improve the recognition performance by augmenting the training dataset with image transformations. The recognition performance of the feature extraction models is compared with an ensemble of fine-tuned networks. The results show that, in the case where long training time is not desirable or a large amount of data is not available, the features from pre-trained DNNs can be used with a shallow classifier to give a comparable recognition accuracy to the fine-tuned networks.

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my committee chair, Prof. Lina Karam for providing me with the full support to pursue my PhD remotely under her supervision despite all the challenges. Thanks to her insightful guidance, her caring nature and her unwavering help, I was able to complete the requirements of PhD and present this dissertation.

I would like to thank the Electrical Engineering department and the ECEE graduate program chair Prof. Joseph Palais for providing me with the opportunity to work on my PhD research from Dubai where I live and work. I would also like to express my thanks to the ECEE graduate advisor Ms. Toni Mengert for her prompt response to my queries.

My gratitude goes as well to Samuel Dodge for co-authoring with me my first accepted journal paper as part of my PhD research work.

I would like to express my sincere gratitude to my brother Naji and his family for their warm hosting whenever I visit Arizona for my PhD work meetings. Without their support and encouragement, my stay would have been way harder.

My sincere thanks go as well to my parents who believed in me and encouraged me through this long journey. Their prayers gave me the strength and will to complete the journey till the end.

I would like to express special thanks to my husband Habib for his loving support and understanding. His constant encouragement and the compromises he did to make this dissertation possible propelled me through these years. Not to forget my two wonderful sons Fouad and Ghadi whose smiling faces and kind words of support were a real blessing for me.

Finally, I would like to thank God for His grace in me.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

Table Page

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

INTRODUCTION

This chapter presents the motivations behind this work and briefly summarizes the contributions and organization of the dissertation.

1.1 Motivation

Accurate biometrics play a critical role in personal authentication and in forensic and security applications. A useful biometric modality has several desirable characteristics: uniqueness, ease of data collection, and preservation of privacy, among others. Uniqueness ensures that the biometric can be used to uniquely identify a person. Ease of data collection enables the biometric to be used in large scale surveillance applications. Privacy preservation is increasingly important as many subjects may not want their personal identity easily accessible. While the ear biometric meets all of the aforementioned desirable characteristics, the face biometric satisfies all of these requirements except for the privacy preservation.

Despite of its impressive growth, face recognition is still receiving a lot of attention because of its high relevance to biometrics, information security, law enforcement needs and surveillance systems, to name a few. One important challenge that limits the effectiveness of face recognition technology is image quality [12]. In real-world environments, the acquired face image quality varies due to lens resolution, focus, distance, noise, illumination, storage, and transmission, to name a few. This may greatly affect the performance of face recognition algorithms. While face recognition has already achieved a very good performance over large-scale galleries that include traditional scene dependent distortions, such as large pose variations, extreme ambi-

ent illumination [13, 14, 15, 16, 17, 18, 19], and partial occlusions due to obstacles and disguise [18, 19, 20, 21, 22], there still exist many more challenges related to variations in image quality, such as additive noise, blur and block occlusion due to packet loss. These types of distortions, which result from capture, processing, and transmission, are commonly present in images and videos acquired by surveillance cameras and increasingly by mobile handheld devices. In such scenarios, it is very likely that the captured image contains a noisy or a blurred face. Moreover, while transmitting compressed face images over lossy packet networks, it is possible that one or more of the packets will not reach the destination. This type of data packet loss during transmission can result in partial occluding blocks, thereby hiding major facial features.

In face recognition, the Sparse Representation Classifier (SRC) method [23] has proved that it could overcome the challenging scene dependent variations including illumination changes, random pixel corruption, and small-size occlusion/disguise. The SRC method assumes that a test image can be represented by a linear combination of sample images from the same subject, which form the basis elements of an over-complete dictionary, via l_1-minimization. In both cases of localized random pixel corruption and low-level occlusion, the error corrupts only a small fraction of the image pixels and is therefore sparse, which can be handled uniformly within the SRC framework, where the component of the test image arising due to occlusion/corruption is naturally separated from the component arising from the identity of the test subject. The authors of [23] show that the choice of features to represent the samples is not important in the SRC framework and, therefore, they simply use raw image pixels as features. Although the SRC shows promising results for clean images and images affected by sparse noise (random pixel corruption and small-size occlusion/disguise), this work shows that under non-sparse image quality distortions that commonly occur

under real-world conditions (e.g., blur, additive noise and relatively large-size occlusions), the choice of features becomes important. Moreover, the SRC framework does not address such distortions in its design and evaluation.

Although ear biometrics have been recently introduced to the forensics field, many approaches have been developed with the aim to improve ear detection and recognition capabilities for reliable deployment in surveillance and commercial applications [24, 25, 26, 27, 28, 29]. These approaches follow a traditional pipeline of normalization, feature extraction and classification. In these works, the main challenge remains a proper selection of feature descriptors that can be resilient to unconstrained conditions, such as illumination changes, occlusion and quality distortions. More recent works (e.g., [30, 31]) use deep neural networks (DNNs) to end-to-end learn a classifier instead of designing a feature-classifier pipeline. This work explores the use of transfer learning with deep neural networks as feature extractors in the more traditional feature-classifier pipeline approach and compares it to a complete end-to-end system. It should be noted that features from pre-trained DNNs have been used in combination with shallow classifiers for a variety of computer vision tasks [32]. This work shows that features from pre-trained networks achieve a strong baseline for unconstrained ear recognition.

1.2 Contributions

The main contribution of this work is in proposing a solution to face recognition under quality distortions, such as blur, additive noise and block occlusion. Since the SRC performs well under traditional scene-dependent variations, the proposed method consists of improving the SRC framework in the presence of such visual quality distortions. No previous work has evaluated the impact of blur on the SRC classifier. Furthermore, it is known that the SRC is not resilient to large distortion

impairments such as contiguous occlusion or random pixel corruption, as these violate the sparse representation assumption that usually holds for modest levels of occlusion/corruption [23]. This work explores the effect of Gaussian blur, realistic blur resulting from camera shake, additive white noise and block occlusion on the SRC and proposes an improved Augmented SRC-based framework (ASRC) that is more robust to blur and noise for any selected feature by accounting for the blur distortion as part of the dictionary construction. Furthermore, the proposed ASRC framework is extended to target block occlusion due to packet loss by replacing blur distortion with block occlusion as part of the dictionary design. A novel Feature Sparse Coding and Classification Index (FSCCI) is proposed to assess both the sparse coding as well as the classification performance of the considered features. This work also proposes a feature selection method based on the FSCCI to better harness the discriminative ability of features when used within the SRC and ASRC frameworks. Rigorous experimentation is conducted on three face recognition benchmarks, namely the ORL [33], Extended Yale B [2], and Labeled Faces in the Wild (LFW) [3] after adding blur, white noise and block occlusion separately to the images at different distortion levels. The obtained results show that the proposed ASRC performs better than SRC at medium to high blur, noise and occlusion levels and that it performs better than other state-of-the-art sparse representation based classification methods [20, 18, 22] and blur invariant methods [34, 35, 36]. The robustness of the proposed ASRC to unseen distortions is also demonstrated by testing its performance in the presence of realistic blur resulting from camera shake.

This work also presents a transfer learning based unconstrained ear recognition method that utilizes existing DNNs pre-trained on the large ImageNet dataset [37] and adapt them for unconstrained ear recognition. The pre-trained feature representations provide a starting point for creating robust classifiers for unconstrained ear

recognition, where they are used to train a shallow classifier. DNN features from five different deep DNN architectures are explored as part of this work: AlexNet [38], VGG16 [39], VGG19 [39], ResNet18 [40], and ResNet50 [40]. The best performance is achieved with the ResNet18 models, which provide consistent performance across the tested datasets.

1.3 Organization

The dissertation is organized as follows. Chapter 2 presents the background in unconstrained face/ear recognition, with a focus on deep learning and sparse-based methods. Chapter 3 introduces a novel Augmented SRC (ASRC) framework for unconstrained face recognition in the presence of blur, occlusion and additive noise distortions. Chapter 4 introduces an unconstrained ear recognition based on deep features extraction. Chapter 5 concludes the dissertation and discusses future work directions.

Chapter 2

BACKGROUND AND RELATED WORK

For years, biometrics played a significant role in forensic science and information security. An increasing number of biometric-based identification systems are being deployed for commercial and safety-oriented applications. Understanding the performance of biometric systems in different real-world environments is key to their application.

A biometric recognition system can be divided into three main stages, namely biometric detection, feature extraction and biometric recognition (Figure 2.1). A biometric recognition system typically starts with detecting the biometric attribute in an image. This step becomes difficult if variations in illumination, position, occlusion and disguise are present. The step of feature extraction is critical for the recognition of the subject's identity, as it consists of computing a robust feature representation for the considered biometric, which can be used to reliably determine the uniqueness of the identity by discriminating between biometrics belonging to different subjects. Finally, the biometric recognition is generally part of either a matching system or an identification system. Matching consists of comparing a biometric with another to approve or reject the matching of identities, while identification compares a biometric with several other given biometrics to find the exact identity among the different subjects.

In this chapter, an overview of the main research methods and issues underlying biometric-based identification systems is given with a focus on ear and face recognition.

Figure 2.1: Biometric recognition scheme. The process applies to various types of biometric attribute including fingerprint, iris, face and ear.

2.1 Biometric Detection Challenges

A recognition system requires the existence of a detection sub-system. The premise is very simple: subjects cannot be recognized if they cannot first be detected. Today, state-of-the-art recognition/detection systems operate relatively robustly in indoor environments with controlled lighting conditions while in outdoor environments, the performance of these systems degrades substantially, mainly due to uncontrolled lighting effects. Current detection and recognition technologies present a challenging problem in the field of image analysis and computer vision in real-world scenarios, and as such a remarkably wide variety of methods have been developed to overcome these challenges. In addition to lighting conditions, there are many equally important factors that cause the appearance of a subject to vary. Examples of such sources of variation follow below:

Age Change: This variation is solely related to face recognition. The shape and texture of the human being face vary with age. Between childhood and teenage, the shape of the skull in addition to the skin texture largely vary resulting in face recognition problems.

Facial Expression: Facial expressions, such as smile, anger, closed eyes/mouth modify the face geometry and texture. Thus, it becomes harder to recognize the identity of the subject.

Lighting Variations: Illumination can be considered a complex problem in both indoor and outdoor recognition. It is well known that lighting changes can cause

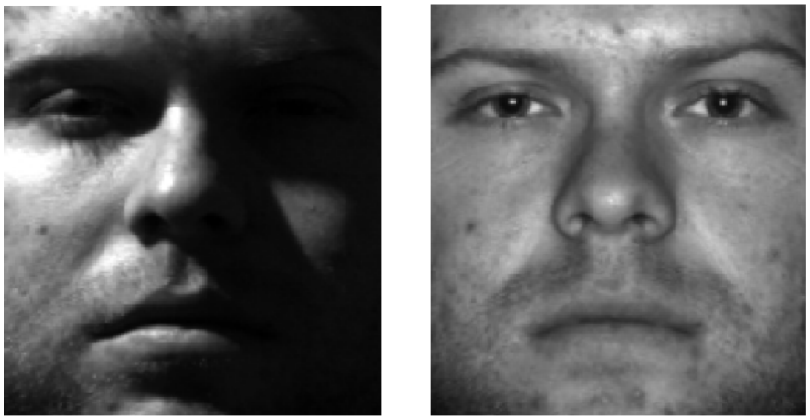


Figure 2.2: The extended Yale B dataset images are affected by extreme illumination variations [2]. The same person seen under different lighting conditions can appear extremely different. In the left image, the light source is from above and to the left; in the right image, the light source is all over the face.

more significant variations of the biometric than those resulting from different subject identities (Fig. 2.2).

Pose Change: Pose variation is mainly due to a rotation out of the plane. The differences in images caused by the change of poses are sometimes larger than the inter-subject images differences. In applications, such as passport control, the images are required to have near frontal poses for the face/ear. However, in uncontrolled environments, like non-intrusive monitoring, faces/ears can be captured in different pose positions, causing a rotation out of the plane.

Presence of Occlusions/Disguise: The use of accessories (sunglasses, scarves, hats, etc.), which partially obstruct the face/ear area, are main factors of occlusion resulting in a loss of information. Moreover, while transmitting compressed face/ear images over lossy packet networks, it is possible that one or more of the packets will not reach the destination. This type of data packet loss during transmission results in partial occluding blocks covering the face/ear.

Blur: Out-of-focus, atmospheric turbulence and relative motion between the sen-

sor and the captured objects represent frequent factors of blur distortion. The blurring process is determined by a point spread function (PSF). A Gaussian distribution function is a special case where the PSF is written as:

$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-} \frac{u^2+v^2}{2\pi\sigma^2}$$

and where (u,v) denotes the pixel location at which the blur PSF is defined. The Gaussian blur PSF accounts for blur due to mainly out-of-focus. For simplicity, the lexicographically ordered $h_\sigma(u,v)$ will be referred to as h_σ, where the blur level is controlled by the variance σ^2. y_b represents the blurred version of y after convolving the blur kernel h_σ with y:

$$y_b = y * h_\sigma \tag{2.1}$$

Noise: Gaussian noise is a common noise type, which is primarily caused by the presence of thermal noise. The latter is inherent in the sensor and arises during capture due to sensor's own temperature in addition to circuitry heating. In many scenarios, a clean image y can be corrupted by additive noise w, where w is a collection of independent identically distributed real-valued random variables following a Gaussian distribution with mean $m = 0$ and variance σ^2 as:

$$y_n = y + w \tag{2.2}$$

where y_n is the noisy test sample.

The Viola Jones detector [41] is a widely used real-time face detector. It has been applied to detect the faces of the Labeled Faces in the Wild (LFW) dataset [3]. The LFW dataset is one of the first benchmark face recognition databases to include a wide variety of real-world facial variations. Some of these variations are displayed in Fig. 2.3.

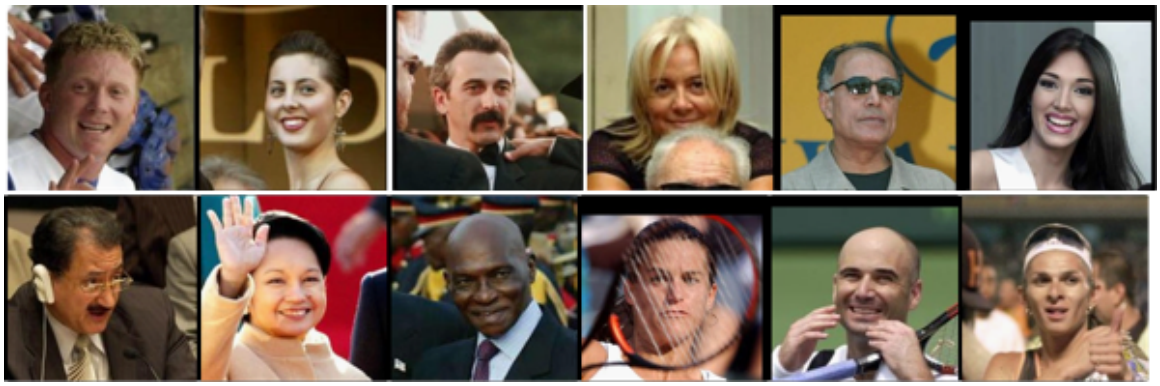


Figure 2.3: Sample images from the unconstrained LFW dataset [3]. Variations in the images include ethnicity, gender, facial expression, pose, occlusion, disguise and lighting.

2.2 Feature Extraction

In the literature, the choice of feature extraction is essential to properly separate object classes in a more discriminative space, also known as feature space. A central issue has been the question of which features are the most important or informative for recognition. The previously-mentioned limitations motivated researchers to deploy feature-based methods that are less sensitive to image variations. These methods are described next and their main advantages and inconveniences are mentioned.

2.2.1 Global Methods

In these approaches, the holistic biometric object is viewed as a vector in the high dimensional image space. This vector is also known as a raw image. Global methods try to find a set of projecting vectors best discriminating different classes. This can be achieved by maximizing the between-class scatter matrix and minimizing the within-class scatter matrix in the projective feature space. These global methods, also known as subspace learning methods, have been widely used with simple classifiers, such as Nearest Neighbor (NN), Nearest Space (NS) and linear Support Vector Machines (SVM). Although these approaches are easy to implement, they are sensitive to image

variations, as any change in the face image results in a change of pixel values.

2.2.1.1 Linear Subspaces

Eigenfaces [4] and Fisherfaces [5] are some examples of popular linear subspace basis vectors that have been widely applied for face recognition under varying illumination and facial expressions changes. The high dimensional data space is projected to a low dimensional feature space using linear projection methods, such as the Principal Component Analysis (PCA) [4] and the Linear Discriminant Analysis (LDA) [5] (Figure 2.4).

Eigenfaces are eigenvectors corresponding to the largest eigenvalues of the covariance matrix computed from the probability distribution over the high-dimensional face space. They form a basis set of all images, such that the original training images are described by a linear combination of the Eigenfaces basis set, producing a dimension reduction that maximizes the total scatter across all classes. If the linear transformation mapping from the original n-dimensional image space $\mathbf{x}_k \, \epsilon \, \mathbb{R}^n$ into the reduced m-dimensional feature space is considered, where $m < n$, then the new feature vectors $\mathbf{y}_k \, \epsilon \, \mathbb{R}^m$ are defined by the linear transformation as follows:

$$\mathbf{y}_k = W^T \mathbf{x}_k, \quad k = 1, 2, ..., N \tag{2.3}$$

where N is the number of sample images and $W^T \, \epsilon \, \mathbb{R}^{m \times n}$ is the linear transformation. If μ is the mean of the input image vectors and if the total scatter matrix is defined by:

$$S_T = \sum_{k=1}^{N} (\mathbf{x}_k - \mu)(\mathbf{x}_k - \mu)^T \tag{2.4}$$

the optimal projection W_{opt} is chosen to maximize the determinant of the total scatter matrix of the projected samples such as: $W_{opt} = arg \; max_W \; |W^T S_T W| =$

$[\mathbf{w}_1\ \mathbf{w}_2\ ...\ \mathbf{w}_m]$, where $\{\mathbf{w}_i|i = 1, 2, ..., m\}$ is the set of n-dimensional eigenvectors of S_T corresponding to the m largest eigenvalues.

While Eigenfaces provide a good representation for the training images, Fisherfaces perform a better classification by improving the scatter to make it more reliable. This method selects W in such a way that the ratio of the between-class scatter and the within class scatter is maximized. The between-class scatter matrix is defined as:

$$S_B = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^T \tag{2.5}$$

and the within-class scatter matrix is defined as:

$$S_W = \sum_{i=1}^{c} \sum_{\mathbf{x}_k \epsilon \mathbf{X}_i} (\mathbf{x}_k - \mu_i)(\mathbf{x}_k - \mu_i)^T \tag{2.6}$$

where μ_i is the mean image of class X_i and N_i is the number of samples in class X_i. W_{opt} is chosen as the matrix which maximizes the ratio of the determinant of the between-class scatter matrix of the projected samples to the determinant of the within-class scatter matrix of the projected samples:

$$\begin{aligned} W_{opt} &= arg\ max_W \frac{|W_T S_B W|}{|W_T S_W W|} \\ &= [\mathbf{w}_1\ \mathbf{w}_2\ ...\ \mathbf{w}_m] \end{aligned} \tag{2.7}$$

where $\{\mathbf{w}_i|i = 1, 2, ..., m\}$ is the set of n-dimensional eigenvectors of S_B and S_W corresponding to the m largest eigenvalues λ_i: $S_B\mathbf{w}_i = \lambda_i S_W \mathbf{w}_i, \quad i = 1, 2, ..., m$.

2.2.1.2 Non-Linear Subspaces

The linearity of the previous subspace methods limits their robustness when applied on complex data, as they are unable to capture non-linear structures. Several non-linear techniques based on the kernel trick have been proposed in the literature, such as K-PCA [42] and K-LDA [43].

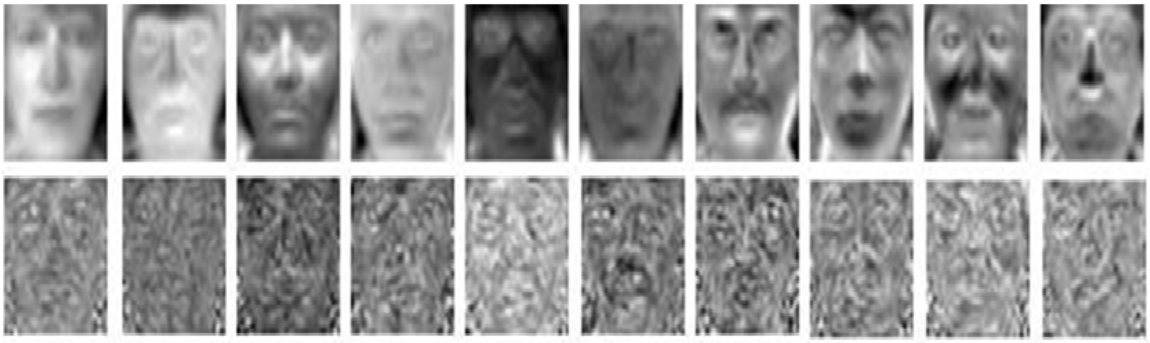


Figure 2.4: Example of subspace learning methods used on the Yale face images. First row corresponds to Eigenfaces [4] and last row corresponds to Fisherfaces [5].

If $\{\mathbf{x}_i | i = 1, 2, ..., n\}$ is a set of n input data samples where $\mathbf{x}_i \, \epsilon \, X$ and X is the input space, a given nonlinear function ϕ maps the input data $\mathbf{x}_i$ to a feature space F with usually very high dimensionality. Let ϕ denote the map from X to F, then:

$$\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle, \quad 1 < i, j < n \tag{2.8}$$

where $\langle , \rangle$ is an inner product in the space F. A kernel based algorithm allows to apply linear methods in the high dimensional space F for the mapped data $\phi(\mathbf{x}_i)$. In other words, the linear algorithms can be recovered by simply using the linear kernel in the kernel based methods.

In the Kernel Principal Component Analysis (K-PCA) algorithm [42], for example, the input data is projected onto the high-dimensional space F by using the nonlinear function ϕ. Then, the standard PCA is performed on the projected space after applying ϕ. The covariance matrix S_T^ϕ and the test sample $\mathbf{y}$ in the projected space F are computed via the kernel function:

$$S_T^\phi = \sum_{i=1}^{N} (\phi(\mathbf{x}_i) - \mu^\phi)(\phi(\mathbf{x}_i) - \mu^\phi)^T \tag{2.9}$$

$$\mathbf{y} = P^T \phi(\mathbf{x}) \tag{2.10}$$

where μ^ϕ is the mean of the transformed input samples $\phi(\mathbf{x}_i)$, $i = 1, 2, ..., N$, and P^T is the set of the eigenvectors of S_T^ϕ corresponding to the m largest eigenvalues.

2.2.1.3 Moment Invariants

Moment invariants were introduced by Hu [44] who derived his seven famous invariants to rotation of 2-D vectors. Since then, numerous works have been deployed to improve Hu's invariants and to apply them in many image analysis applications [45, 46, 47, 48, 49].

Moments are essentially known to be of two main types, geometric and complex. A geometric moment m_{pq} of an image $f(x, y)$ and of order $p + q$, where p and q are non-negative integers, is defined as:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \tag{2.11}$$

The corresponding central moment μ_{pq} and the normalized moment ν_{pq} are defined as:

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - x_c)^p (y - y_c)^q f(x, y) dx dy \tag{2.12}$$

$$\nu_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\omega}} \tag{2.13}$$

where the coordinates (x_c, y_c) correspond to the centroid of $f(x, y)$ and $\omega = (p + q + 2)/2$. The complex moment c_{pq} of the image $f(x, y)$ in its turn is defined as:

$$c_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x + iy)^p (x - iy)^q f(x, y) dx dy \tag{2.14}$$

where i denotes the imaginary unit. The corresponding central and normalized moments are defined as in (2.12) and (2.13). Hu [44] published seven rotation invariants consisting of second and third order moments, where the first four moments invariants

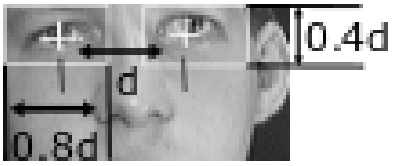


Figure 2.5: Example of eyes width measurement [6]. The measurement between the rough eyes positions (circles) allow to compute the size and location of the window that will locate the accurate central points of the eyes (crosses).

are:

$$\begin{aligned}
\phi_1 &= \mu_{20} + \mu_{02} \\
\phi_2 &= (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \\
\phi_3 &= (\mu_{30} - \mu_{12})^2 + 3(\mu_{21} - \mu_{03})^2 \\
\phi_4 &= (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2
\end{aligned} \tag{2.15}$$

Other moment invariants have been proposed, such as invariants to affine transform [50, 51] and convolution [45, 47, 48, 49, 34].

2.2.2 Local Methods

These methods are known as local approaches because they extract local features at specific regions instead of considering the holistic biometric attribute. This type of features, also known as hand-crafted features, can be classified into two main categories as described next.

2.2.2.1 Interest Point Features

These methods rely on locating interest points or keypoints in each image, and calculating a feature description from the pixel region surrounding the interest point. Keypoints may include corners, edges or contours, and larger features or regions such as blobs. Some of these methods use directly the face/ear characteristic points [52, 28, 7], while the other methods develop more elaborated representations of information carried by the biometric characteristic points, rather than just the geometric

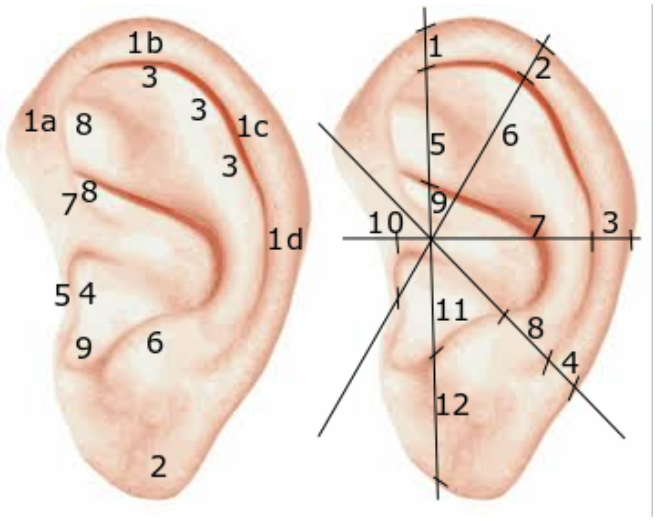


Figure 2.6: The Iannarelli ear system [7]. (a) Anatomy, (b) Measurements.

characteristics [53, 54, 55, 56, 57, 58, 59]. For face recognition, geometric features, such as the width of the head, the distance between the eyes (Figure 2.5), etc., are extracted [6, 60, 61]. For ear recognition, the Iannarelli System of Ear Identification [7] is an example where geometric-based measurements around the ear are computed for a unique ear characterization (Fig. 2.6). Interest point features can be effectively used for recognition where only one reference image is available. However, their performance depends on many effective detectors for locating facial feature points. In practice, detecting an accurate characteristic point is not easy, especially in cases where the shape or appearance of a facial image can vary widely [62].

2.2.2.2 Feature Descriptors

The image is divided into small regions (or patches) where local characteristics are computed and extracted. Alternatively, patches are taken around detected interest points. The vectors that are generated to describe these characteristics or features are called descriptors. The commonly-used feature descriptors are: Gabor coefficients [52], Haar wavelets [41], Fourier transforms, scale-invariant feature transform (SIFT)

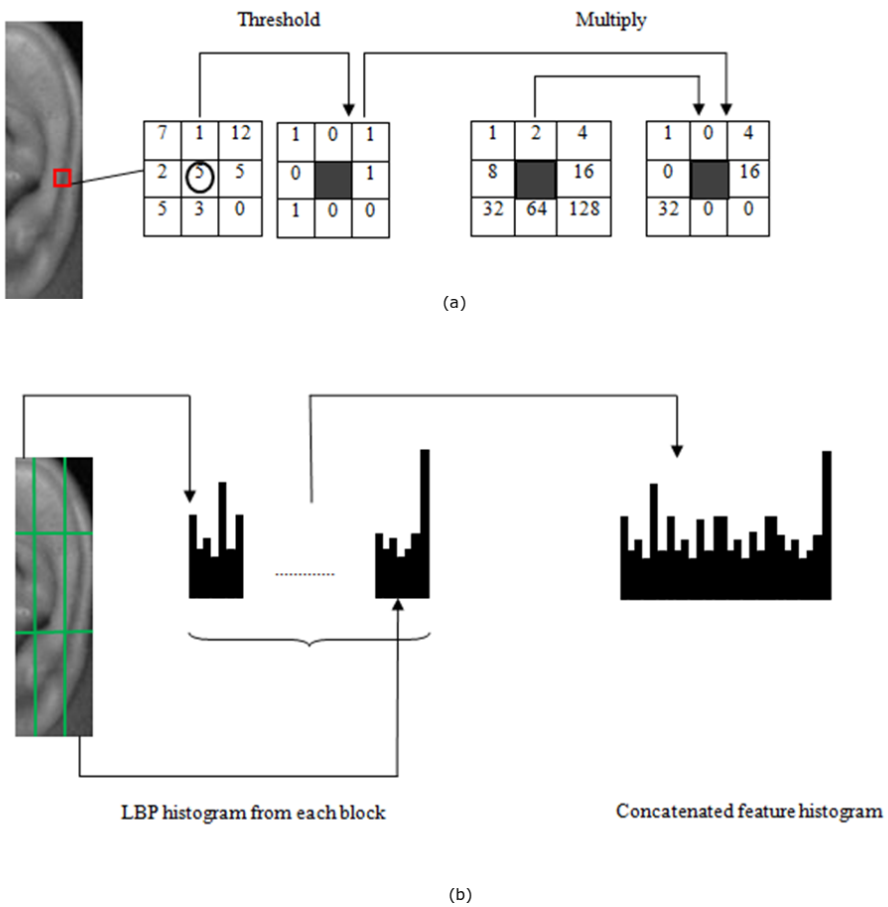


Figure 2.7: The LBP feature descriptor: (a) the basic LBP operator applied to one pixel in the ear image; (b) the resulting LBP concatenated feature histogram.

[63], local binary pattern (LBP) [64], histogram of oriented gradients (HOG) [65], local phase quantization (LPQ) [66] and binarized statistical image features (BSIF) [67].

Some of these descriptors represent features as binary bit vectors [64, 68, 67]. To compute the features, pairs of image pixels are compared and the results are stored as binary values in a vector. The LBP [64], for instance, creates a descriptor or texture model using a set of histograms of the local texture neighborhood surrounding each pixel. In this case, local texture is the feature descriptor, which makes the LBP a computationally simple texture metric. The basic LBP operator thresholds the 3×3

neighborhood of each pixel with the center value and represents the result as a binary code. The histogram of these binary values is then derived to compute the LBP feature descriptor. A uniform LBP, which contains at most two bitwise transitions, accounts for most of the patterns in object recognition. The 58 different uniform patterns are assigned to different bins when generating the histogram, while all the non-uniform patterns are assigned to one bin. Therefore, the dimension of the LBP descriptor is 59. Usually, the image is subdivided into blocks or local regions where the LBP histograms are computed and then concatenated into one representative feature descriptor. Fig. 2.7 illustrates the LBP computation for an ear recognition application. Local binary pattern methods achieve very good accuracy and robustness compared to other methods.

The other descriptors, also called spectral descriptors, involve more intense computations and algorithms, as they measure light intensity, color, local area gradients, local area statistical features and moments, surface normals, and histograms of local gradient direction. A spatial-frequency analysis is often desirable to extract local features that are robust against image variations and distortions.

Among various wavelet bases, Gabor wavelets are popular for measuring local spatial frequencies. A family of Gabor wavelets of different orientations and frequencies are applied to the image. The magnitudes of the Gabor wavelet coefficients at each location in the image are used for feature representation.

The SIFT [63], which is the most well-known method for finding interest points and feature descriptors, provides invariance to scale, rotation, illumination, affine distortion, perspective and similarity transforms, and noise.

The HoG method [65] is commonly used for image classification, and relies on computing local region gradients over a dense grid of overlapping blocks. Fig. 2.8 illustrates the different steps of computing the HoG descriptor for an ear recognition

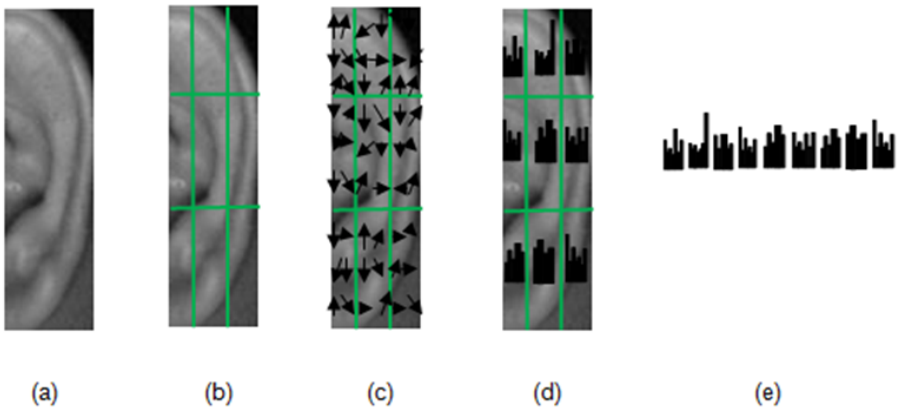


Figure 2.8: The HoG feature descriptor. (a) The ear image. (b) The subdivided ear image into blocks. (c) The gradient orientations. (d) The histogram of gradient orientations at every block. (e) The concatenation of histograms.

application. HoG descriptors can be found in two main variants, the Dalal *et al.* [65] original variant and the slightly improved Felzenszwalb *et al.* [69] UIUC variant. The detection window includes overlapping blocks of size 16×16 with a stride of 8×8 pixels. Each block is composed of 2×2 cells where the cell size is 8×8 pixels. The original variant of HoG computes a 36-dimensional feature vector that accounts for a linear gradient voting into 9 undirected orientation bins. On the other hand, the adopted UIUC variant of HoG computes the directed gradients as well as a four dimensional texture-energy feature for each cell. The histogram of orientations, which includes 9 bins for the undirected orientations, and the four-dimensional texture-energy are augmented with both contrast sensitive and contrast insensitive features, leading to a 31-dimensional feature vector. Gradient magnitude histogram values are normalized to unit length to provide illumination invariance. This variant [69] accounts more than the original one [65] for rotations and translations by considering directed orientations. It also includes the texture-energy feature, which improves the performance in capturing local information in the images.

The LPQ descriptor [66] was designed to be robust to image blur, and it leverages

the blur insensitive property of Fourier phase information. LPQ is reported to provide robustness for uniform blur, as well as uniform illumination changes. It also provides equal or slightly better accuracy on non-blurred images than LBP and Gabor filter bank methods. While mainly used for texture description, LPQ can also be used for local feature description to add blur invariance by combining LPQ with another descriptor method such as SIFT.

Haar features became popular in the field of computer vision by the Viola Jones [41] algorithm. They are based on specific sets of rectangle patterns, which approximate the basic Haar wavelets by averaging the pixel values within the rectangle. This is efficiently computed using integral images. However, Haar features have drawbacks, since rectangles by nature are not rotation invariant for angles beyond 15 degrees. Also, the integration of pixel values within the rectangle destroys fine detail.

Feature descriptors have proven to work well for recognition applications in constrained environments.

2.2.3 Deep Learning Methods

More recently, deep learning methods, specifically convolutional neural networks (CNN), also known as deep neural networks (DNNs), have achieved high recognition accuracies in the field of computer vision. The architecture of these DNNs is described next and their different advantages and drawbacks are indicated.

AlexNet [38] was the first deep neural network to achieve success on the large scale ImageNet dataset [37]. The model architecture, which has 60 million parameters and 500,000 neurons, consists of five convolutional layers and three fully connected layers with a final 1000-way softmax.

The VGG networks [39] extend the AlexNet framework by adding more layers between the pooling stages. Compared with AlexNet, a single convolutional layer

between pooling stages is replaced with multiple stacked convolutional layers, which are followed by three fully connected layers. The final layer is the softmax layer. The VGG style networks, which include 133 million to 144 million parameters, use small 3×3 size filters to reduce the number of parameters and consequently reduce overfitting. There are many variants of VGG numbers of layers, with the most popular variants being the 16 layer VGG16 model and the 19 layer VGG19 model.

As the networks become deeper, the gradients can vanish (or explode). ResNet [40] networks use "skip" connections between convolutional blocks in order to create much deeper neural networks while ensuring that there is no vanishing (or exploding) gradient problem. The layers are formulated as learning residual functions with respect to the layer inputs, instead of learning more simple feed-forward functions. Despite of their large depth, ResNets have much less number of parameters varying between 11.7 million (18 layers) and 60.2 million (152 layers).

Deep neural networks evolved to meet the requirements of unconstrained face recognition. The literature proposed several high performing face deep networks that were almost able to approach humans on challenging face benchmarks such as LFW [3].

DeepFace uses a deep CNN trained to classify faces using a dataset of 4 million images corresponding to 4000 different identities. It also uses a siamese network architecture, where the same CNN is applied to pairs of faces to obtain descriptors that are then compared using the Euclidean distance. The goal of training is to minimise the distance between pairs of faces portraying the same identity and maximise the distance between pairs that belong to different identities. In addition to using a very large amount of training data, DeepFace uses an ensemble of CNNs, as well as a pre-processing phase in which face images are frontalized using a 3D model.

DeepID is referred to as Deep hidden IDentity (DeepID). Unlike DeepFace whose

layer	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
type	input	conv	relu	conv	relu	mpool	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	relu	mpool	conv
name	-	conv1_1	relu1_1	conv1_2	relu1_2	pool1	conv2_1	relu2_1	conv2_2	relu2_2	pool2	conv3_1	relu3_1	conv3_2	relu3_2	conv3_3	relu3_3	pool3	conv4_1
support	-	3	1	3	1	2	3	1	3	1	2	3	1	3	1	3	1	2	3
filt dim	-	3	-	64	-	-	64	-	128	-	-	128	-	256	-	256	-	-	256
num filts	-	64	-	64	-	-	128	-	128	-	-	256	-	256	-	256	-	-	512
stride	-	1	1	1	1	2	1	1	1	1	2	1	1	1	1	1	1	2	1
pad	-	1	0	1	0	0	1	0	1	0	0	1	0	1	0	1	0	0	1

layer	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
type	relu	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	softmax
name	relu4_1	conv4_2	relu4_2	conv4_3	relu4_3	pool4	conv5_1	relu5_1	conv5_2	relu5_2	conv5_3	relu5_3	pool5	fc6	relu6	fc7	relu7	fc8	prob
support	1	3	1	3	1	2	3	1	3	1	3	1	2	7	1	1	1	1	1
filt dim	-	512	-	512	-	-	512	-	512	-	512	-	-	512	-	4096	-	4096	-
num filts	-	512	-	512	-	-	512	-	512	-	512	-	-	4096	-	4096	-	2622	-
stride	1	1	1	1	1	2	1	1	1	1	1	1	2	1	1	1	1	1	1
pad	0	1	0	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0

Figure 2.9: Details of the VGG-Face deep neural network architecture A as described in [8].

features are learned by one single big CNN, DeepID is learned by training an ensemble of small CNNs, used for network fusion. The patches of facial images are input to every single CNN and the features learned by all CNNs are concatenated to form one powerful feature descriptor. Both RGB and greyscale patches, which are extracted around facial points, are used to train the DeepID. The length of the DeepID feature is 2 (RGB and Greyscale) $\times$ 60 (patches) $\times$ 160 (feature length of one network) = 19,200. Each network consists of 4 convolutional layers, 3 max pooling layers and 2 fully connected layers.

The VGG-Face CNN architecture A, as described in Parkhi *et al.* [8], is shown in full detail in Figure 2.9. It is based on the VGG-Very-Deep-16 CNN architecture. It comprises 11 blocks, each containing a linear operator followed by one or more non-linearities such as ReLU and maxpooling. The first eight blocks are convolutional, as the linear operator is a bank of linear convolution filters. The last three blocks are Fully Connected (FC) in the sense that they are also convolutional, but the size of the filters matches the size of the input data, such that each filter reads data from the entire image. The first two FC layers output are 4,096 dimensional and the last FC layer has either N = 2,622 or N = 1,024 dimensions, depending upon the loss functions used for optimisation, either N-way class prediction.

The FaceNet architecture, as described in Figure 2.10, is designed by Google

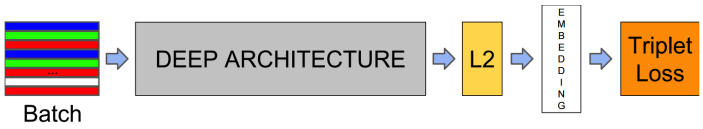


Figure 2.10: Outline of the FaceNet deep neural network architecture as described in [9]. This network consists of a batch input, a deep CNN followed by l_2 normalization, and finally a triplet loss during training.

researchers and consists of convolutional layers that are inspired from GoogLeNet inception models. The FaceNet returns a 128 dimensional vector embedding for each face. Having been trained with triplet loss to reinforce the similarity of images belonging to the same identity and the difference of images corresponding to different subjects, the 128 dimensional embedding can effectively cluster faces. Hence, the embedding vectors would be closer for similar faces and more distant for dissimilar faces. The FaceNet architecture is trained over a dataset with a very large number of labeled faces belonging to numerous subjects and including different variational conditions, such as pose and illumination.

SphereFace [70] is the latest state-of-the-art in deep face recognition. The authors in [70] propose the angular softmax (A-Softmax) loss that enables convolutional neural networks (CNNs) to learn angularly discriminative features. Geometrically, A-Softmax loss can be viewed as imposing discriminative constraints on a hypersphere manifold, which intrinsically matches the prior that faces also lie on a manifold. Moreover, the size of the angular margin can be quantitatively adjusted by a parameter m.

In their original work, 'deep' methods have only been evaluated on clean images that do not include any visual quality distortions, such as blur, noise, compression and contrast. Despite of their excellent performance on sharp undistorted images, deep learning methods do not perform well in the presence of visual distortions. Dodge and Karam [71] provide an evaluation of state-of-the-art DNN models for

image classification under quality distortions where they show that the performance of these DNNs is significantly reduced in the presence of blur, noise, contrast, JPEG, and JPEG2000 compression. Grm *et al.* show equally in their work [72] that high levels of noise, blur, missing pixels, and brightness have a detrimental effect on the verification performance of deep models, such as AlexNet [38], VGG-Face [8], GoogleNet [73] and SqueezeNet [74].

2.3 Recognition Methods

Recognition can be viewed as classifying the probe images into identifiable classes via the extraction of significant features of the biometric attributes.

Normally, the recognition process makes use of one of the following two classification strategies: i. Supervised classification in which the probe image is identified as a member of a predefined class. ii. Unsupervised classification (clustering) in which the image is assigned to an unknown class.

The well-known approaches that are widely used to solve pattern recognition problems including clustering technique (k-means algorithm), statistical pattern classifiers (k-nearest neighbour classifier and Bayesian classifier) and ensemble learning classifiers (bagging, boosting and stacking) are equally used for recognizing face/ear patterns. There will be a focus in this section on two supervised classifiers that are of high use in this work (Chapter 3 and Chapter 4), the support vector machines (SVM) and the sparse representation classifier (SRC).

2.3.1 Support Vector Machines

Support vector machines (SVMs) [75] are powerful tools of classification in the machine learning community. Initially, SVMs are linear classifiers that are designed to support binary classification problems. Later, the one-against-one technique is

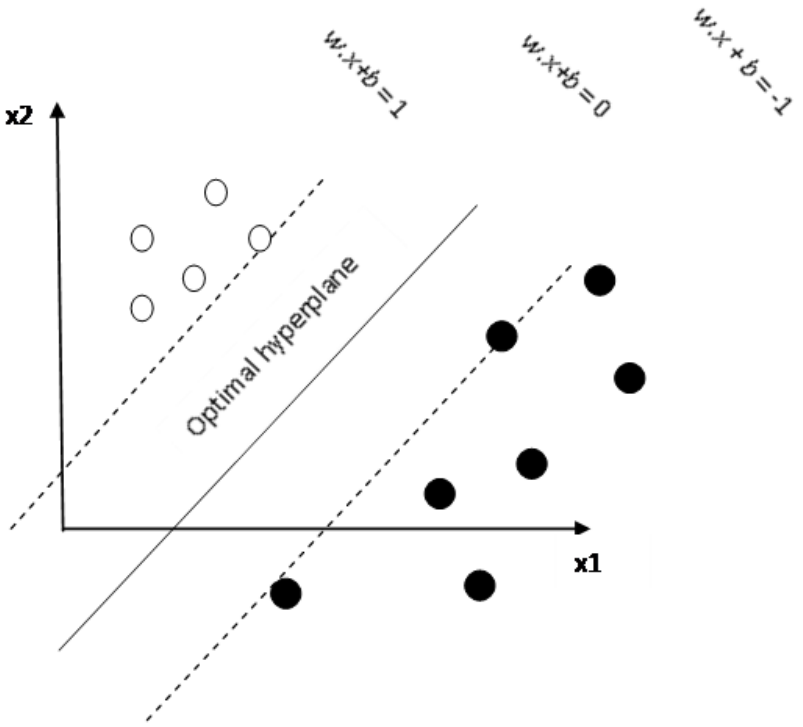


Figure 2.11: SVM binary classification around the optimal hyperplane where the margin is maximal.

used for mutli-class classification. It fits all binary subclassiers and finds the correct class by a voting mechanism. The binary classification approach will be described next.

Suppose that a linearly separable set of training samples $\{S = (\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$ exists, where $\mathbf{x} \epsilon \mathbb{R}^{d \times n}$ denotes the input space that includes input sample vectors of dimension d each, and $\mathbf{y} = \{-1, +1\}$ is the output space indicating the class of binary classification. The linear SVM computes an optimal linear hyperplane that separates between the two classes by maximizing the margin between the classes closest points (Fig. 2.11). The points lying on the boundaries are called support vectors while the middle of the margin is the optimal separating hyperplane. The points $\mathbf{x}$, which lie on the hyperplane satisfy: $w\mathbf{x} + b = 0$, where w defines a direction perpendicular to the hyperplane, while varying the value of b moves the hyperplane parallel to itself.

Usually, two parallel hyperplanes are selected to separate the two classes of data, such that the distance between them is as large as possible. The region bounded by the two hyperplanes is called the margin. The sample points that are located on the wrong side of the separating hyperplane are weighted down to reduce their influence on the classification. The hinge loss function is used for this purpose:

$$max(0, 1 - y_i(w\mathbf{x}_i + b)), \quad 1 \leq i \leq n \tag{2.16}$$

For the samples on the wrong side of the hyperplane, the function's value is proportional to the distance from the margin. Thus, the SVM classification is implemented by minimizing the following equation:

$$\frac{1}{n}\sum_{i=1}^{n} max(0, 1 - y_i(w\mathbf{x}_i + b)) + \lambda||w||_2^2 \tag{2.17}$$

where λ determines the tradeoff between increasing the margin-size and ensuring that the samples points lie on the correct side of the margin. If a linear separating hyperplane cannot be found, sample points are projected into a high dimensional space F where the sample points can become linearly separable:

$$\phi : \mathbb{R}^d \to F \quad \phi : x_i \to \phi(x_i) \tag{2.18}$$

ϕ is a non-linear mapping where the dot product $\langle ., . \rangle$ is defined in F by: $\langle \phi(x_i), \phi(x_j) \rangle = \mathbf{K}(x_i, x_j)$. $\mathbf{K}$ is a kernel function that represents the data in the reproducing kernel Hilbert space where they can be linearly separable.

2.3.2 Sparse Representation Classifier (SRC)

The Sparse Representation Classifier (SRC) method [23] has proved that it could overcome the challenging scene dependent variations including illumination changes, random pixel corruption, small size face occlusion/disguise. The SRC method assumes that a test image can be represented by a linear combination of sample images

from the same subject, which form the basis elements of an overcomplete dictionary, via l_1-minimization. The authors of [23] show that the choice of features to represent the samples is not important in the SRC framework and, therefore, they simply use raw image pixels as features.

Consider a set of training samples corresponding to M object classes with m_i samples $\{D_1^i, D_2^i, \cdots, D_{m_i}^i\} \epsilon \mathbb{R}^{d \times m_i}$in the i^{th} object class, $1 \leq \mathrm{i} \leq M$. The dictionary D is formed by inserting the training samples D_j^i of all M object classes as entries, resulting in:

$$D = \{D_1^1, \cdots, D_{m_1}^1 | \cdots | D_1^M, \cdots, D_{m_M}^M\} \tag{2.19}$$

Given a sufficient number of training samples in the i^{th} object class, a test sample vector $y_t \epsilon \mathbb{R}^{d \times 1}$ from the same class can be represented as a sparse linear combination of the training samples in D as follows:

$$y_t = D.\alpha^* \tag{2.20}$$

where $\alpha^* \epsilon \mathbb{R}^{K \times 1}$ is the sparse coefficient vector and $K = \sum_{i=1}^{M} m_i$ is the number of training samples in D. The sparse vector α^* is computed by solving the following constrained l_1-norm minimization problem:

$$\alpha^* = arg\ min_\alpha \ \ ||y_t - D.\alpha||_2^2 \ + \lambda ||\alpha||_1 \tag{2.21}$$

where $||.||_1$ and $||.||_2$ denote the l_1-norm and l_2-norm respectively, and $0 \leq \lambda \leq 1$ is a sparsity coefficient.

Once (2.21) is solved, the representation residual for the i^{th} object class is computed as follows:

$$r_i = ||y_t - D.\delta_i(\alpha^*)||_2^2 \tag{2.22}$$

where $\delta_i(\alpha^*)\ \epsilon\ \mathbb{R}^{K\times 1}$ is a vector whose non-zero entries are the entries in α^* that are associated with the i^{th} class.

The test sample y_t will be assigned to the class i^* corresponding to the minimum representation residual:

$$i^* = \arg\ \min_i(r_i),\ i = 1, ..., M \tag{2.23}$$

The authors in [23] try to understand geometrically how the choice of the feature affects the ability of the l_1-minimization to recover the correct sparse solution α^*. If P_γ denotes the l_1-ball of radius γ:

$$P_\gamma = \{\alpha : ||\alpha||_1 \leq \gamma\} \subset \mathbb{R}^K \tag{2.24}$$

The unit l_1-ball P_1 is mapped to the polytope $P = D(P_1) \subset R^d$ consisting of all y that satisfy $y = D\alpha$ for some α whose l_1-norm is ≤ 1. Geometrically, finding the minimum l_1-norm solution α^* is equivalent to expanding the l_1-ball P_γ until the polytope $D(P_\gamma)$ first touches y. The value of γ at which this occurs is exactly $||\alpha^*||_1$. If D maps all t-dimensional facets of P_1 to facets of P, the polytope P is referred to as t-neighborly [76]. The neighborliness of the polytope P increases with the feature dimension d. Although the most data-dependent features popular in face recognition might give highly neighborly polytopes P, the authors in [23] reveal that if the solution α^* is sparse enough, it can be correctly recovered via l_1-minimization from any sufficiently large number d of linear measurements. If α^* has $t << K$ nonzeros, the minimum required value of d is given as:

$$d \geq 2t \log(K/d) \tag{2.25}$$

where K is the total number of training samples.

Chapter 3

AUGMENTED SPARSE CLASSIFIER (ASRC) FOR FACE RECOGNITION UNDER QUALITY DISTORTIONS

In the last two decades, numerous methods have been developed to offer a formulation to the face recognition problem under scene-dependent conditions, such as illumination/pose variations, random pixel corruption and disguise. However, these methods have not considered image quality degradations resulting from capture, processing and transmission, such as blur, additive noise and occlusion due to packet loss, under the same scene variations. Although deep neural networks are achieving state-of-the-art results on face recognition, the existing networks are susceptible to quality distortions. In this work, the performance of a well-known face recognition framework, namely the sparse representation classifier (SRC), is explored in the presence of blur, additive noise, and block occlusions, in both constrained and unconstrained environments. The SRC has shown a good performance in the presence of different interclass variations. While the SRC was shown to be a framework independent of the extracted features for sparse representation, this work shows that feature extraction within SRC is important when quality distortions are present. To this end, a Feature Sparse Coding and Classification Index (FSCCI) is presented in this work, such that it is capable of assessing the quality of features in terms of recognition accuracy while preserving the sparsity of the representation. In the evaluation of the SRC framework, three main types of features are considered including image raw pixels, HoG and deep learning VGG-Face. Next, an Augmented SRC (ASRC) framework is proposed to improve the performance of the original SRC in the presence of Gaussian blur, while preserving its robustness to scene dependent variations.

It is further proposed to apply the ASRC framework to recognize faces that have been corrupted by block occlusion and additive noise, as the SRC has proved that it could only overcome a limited level of face occlusion/corruption. Additionally, the robustness of the model to unseen real-world distortions, such as camera shake blur, is demonstrated. The obtained performance results show that the proposed method outperforms state-of-the-art sparse-based methods, including SRC, and blur invariant methods.

3.1 Introduction

Face recognition is still receiving a lot of attention because of its high relevance to biometrics, information security, law enforcement needs and surveillance systems, to name a few. One important challenge that limits the effectiveness of face recognition technology is image quality [12]. In real-world environments, the acquired face image quality varies due to lens resolution, focus, distance, noise, illumination, storage, and transmission, to name a few. This may greatly affect the performance of face recognition algorithms. While face recognition has already achieved a very good performance over large-scale galleries that include traditional scene-dependent distortions, such as pose variations, extreme ambient illumination [13, 14, 15, 16, 17, 19], and partial occlusions due to obstacles and disguise [23, 19, 18, 22, 21, 20], there still exist many more challenges related to variations in image quality, such as blur, additive noise, and block occlusions due to packet loss. These types of distortions, which result from capture, processing, and transmission, are commonly present in images and videos acquired by surveillance cameras and increasingly by mobile handheld devices or transmitted over IP and wireless networks. In such scenarios, it is very likely that the captured/transmitted image contains a noisy or a blurred face. Moreover, while transmitting compressed face images over lossy packet networks, it is

Table 3.1: Main Factors in Capture and Transmission for Image Quality Degradation.

Noise	Blur	Block Occlusion
Thermal	Out of focus	Transmission errors/loss
Shot	Motion (camera/subject)	Low device performance
Quantization	Shallow depth of field	Software issues
High ISO/long exposure	Turbulent medium (fog/rain)	Faulty hardware

possible that one or more of the packets will not reach the destination. This type of data packet loss during transmission results in partial occluding blocks, thereby hiding major facial features. Other common reasons for occlusion are scene-dependent obstacles and disguise accessories that cover one or more random regions in the face. Table 3.1 lists some of the main factors for image quality degradation during capture and transmission.

Sparse representations have shown huge potential capabilities in handling problems such as image denoising [77, 78, 79, 80, 81, 82, 83, 84]. The literature has shown that the need for sparse representations may arise when noise exists in image data since sparse representations extract the sparse image components, which are regarded as useful information, and disregard the representation residual, which is considered as the image noise term. Finally, the image can be reconstructed by employing the obtained sparse components resulting in a noise-free image [78, 79, 80, 81]. In face recognition, the Sparse Representation Classifier (SRC) method [23] has proved that it could overcome challenging scene-dependent variations including illumination changes, random pixel corruption, and small-size occlusion/disguise. The SRC method assumes that a test image can be represented by a linear combination of sample images from the same subject, which form the basis elements of an overcomplete dictionary, via l_1-minimization. In both cases of random pixel corruption and low-

level occlusion, the error corrupts only a fraction of the image pixels and is therefore sparse, which can be handled uniformly within the SRC framework, where the component of the test image arising due to occlusion/corruption is naturally separated from the component arising from the identity of the test subject. The authors of [23] show that the choice of features to represent the samples is not important in the SRC framework and, therefore, they simply use raw image pixels as features. Although the SRC shows promising results for clean images and images affected by sparse noise (random pixel corruption and small-size occlusion/disguise), it is argued and shown in this work that under non-sparse image quality distortions that commonly occur under real-world conditions (e.g., blur, additive noise and relatively large-size occlusions), the choice of features becomes important.

Numerous sparse representation-based classification methods followed later with the sole aim of improving the robustness of the SRC method to appearance variations, such as the extended sparse representation classifier (ESRC) [22], robust sparse coding (RSC) [18], and structurally incoherent low-rank matrix decomposition (LRSI) [20]. Although the above methods show promising results with images affected by sparse noise, they have not considered acquisition and transmission distortions, such as blur and non-sparse noise (additive noise and relatively large-size occlusions), as part of their design. Furthermore, these methods require that the training images be well aligned for reconstruction purposes. However, the alignment methods involve apriori knowledge of facial landmarks, which become inaccurate when the face quality is degraded with blur and non-sparse noise, as subtle features will become masked while other misleading ones will be introduced.

Blur invariant methods have been proposed to reduce the sensitivity of images to blur distortion. Several authors proposed different blur invariants that were mainly based on image moments without the need of image restoration. Flusser *et al.* [47]

and Flusser and Suk [45] proposed a system of blur invariants, which are recursive functions of geometric moments of the image and proved their invariance under a convolution with arbitrary centrosymmetric kernel. Flusser and Zitova [46] developed further the concept of blur invariants by introducing combined invariants to convolution and rotation that were successfully used in satellite image registration. Zhang *et al.* [85] and Suk and Flusser [86] proposed combined invariants to convolution and affine transform. These invariants were used in different non-face recognition applications, such as aircraft silhouette recognition and sign language recognition. Other researchers developed blur invariants that are based on orthogonal moments instead of geometric moments. Legendre moments [48, 87, 88, 89], Zernike moments [90, 91, 92], and Chebyshev moments [93] are some examples. Some other authors proposed blur invariants in Fourier domain. Ojansivu and Heikilla [94, 95] and Tang *et al.* [96] used blur-invariant properties of Fourier transform phase for image registration and matching. Later, Pedone *et al.* [97, 98] generalized the same idea. The popular method of local phase quantization (LPQ) [66] also belongs to the same category. Although the Gaussian kernel is a special case of symmetric kernels, these blur invariant methods do not work well with Gaussian blur. Few attempts to derive invariants to Gaussian blur have been reported. Xiao *et al.* [99] derived invariants to Gaussian blur but did not use the Gaussian form explicitly. Instead, they used the circular symmetry property. Gopalan *et al.* [100] derived blur invariants but did not make any assumption on the parametric shape of the kernel. Zhang *et al.* [34] derived a blur invariant distance that is specifically designed for Gaussian blur. Although the blur invariants are not explicitly defined, the invariant distance measure was used for object classification. Flusser *et al.* [35] developed new invariants to Gaussian blur for face recognition based on the same concept of Zhang's distance [34]. These invariants are based on projection operators in the Fourier domain and on image moments in

the image domain. The authors showed that the proposed invariants outperformed the blur-invariant Zhang's distance [49, 34] and the local phase quantization (LPQ) blur-invariant features [66]. Nevertheless, the method in [35] is not a good design for blurred face recognition as the authors have assumed the use of the same images for training and testing with the latter being blurred. On the other hand, Vageeswaran *et al.* [36] proposed a blur-robust algorithm (rDRBF) for unconstrained blurred face images, which showed a significant improvement over the LPQ algorithm of [66] for large levels of Gaussian blur. The method in [36] solves a convex l_1-norm problem to create artificially blurred versions of the gallery images where the blurred probe image is matched to them. A major constraint of the latter method is that it applies blur estimation for every gallery image, which is computationally complex.

More recently, deep learning methods, specifically convolutional neural networks (CNN), also known as deep neural networks (DNNs), have achieved high accuracies in face recognition, such as DeepFace [101], DeepID [102], VGG-Face [8], and FaceNet [9]. Despite their excellent performance, 'deep' methods are susceptible to visual distortions. Dodge and Karam [71] provide an evaluation of state-of-the-art DNN models for image classification under quality distortions where they show that the performance of these DNNs is significantly reduced in the presence of blur and noise distortions. Grm *et al.* show equally in their work [72] that high levels of noise and blur have detremental effect on DNNs such as AlexNet [38], VGG-Face [8], GoogleNet [73] and SqueezeNet [74]. Fernandez studies in [103] the performance of state-of-the-art DNNs on objects that have been occluded by different block sizes. The author shows that the performance of theses networks decreases sharply in the presence of an increasing occlusion size.

The main contribution of this work is in proposing a solution to face recognition under quality distortions, such as blur, non-sparse additive noise and relatively large-

size block occlusions. Since the SRC performs well under traditional scene-dependent variations, the proposed method consists of improving the SRC framework in the presence of such visual quality distortions. No previous work has evaluated the impact of blur on the SRC classifier. Furthermore, it is known that the SRC is not resilient to large distortion impairments such as contiguous occlusion or non-sparse random pixel corruption, as these violate the assumption that the considered images have sparse representation with respect to the extended identity matrix dictionary that usually handles well modest levels of occlusion/corruption [23]. Therefore, as part of this work, the effect of blur, additive noise and occlusion on the SRC is explored and an improved Augmented SRC-based framework (ASRC) is proposed that is more robust to blur and non-sparse noise/occlusion for any selected feature by accounting for these distortions as part of the dictionary construction. Moreover, a novel Feature Sparse Coding and Classification Index (FSCCI) is proposed to assess both the sparse coding as well as the classification performance of the considered features. This work also proposes a feature selection method based on the FSCCI to better harness the discriminative ability of features when used within the SRC and ASRC frameworks. Rigorous experimentation is conducted on three face recognition benchmarks, namely the ORL [33], Extended Yale B [2], and Labeled Faces in the Wild (LFW) [3] after adding Gaussian blur, camera shake blur, white noise and block occlusions separately to the images at different distortion levels. The obtained results show that the proposed ASRC performs better than state-of-the-art sparse representation-based classification methods [20, 18, 22], including the standard SRC [23], and blur invariant methods [34, 35, 36].

The rest of this chapter is organized as follows. Section 3.2 describes the SRC classifier's limitations in the presence of visual distortions. Section 3.3 presents the proposed method in detail. The experimental setup and results are presented in

Section 3.4. Finally, Section 3.5 concludes with some final remarks and discussions.

3.2 SRC Limitations

The SRC framework described in Section 2.3.2 shows that a test sample y can be sufficiently represented using only the training samples from the same class as shown in (2.20) if the representation is naturally sparse. The more sparse the recovered α^* is, the easier will it be to accurately determine the identity of the test sample y. The authors in [23] indicate that the choice of features is not critical enough to affect the sparsity of the representation as long as the feature dimension surpasses a threshold that is predicted by the theory of sparse representation. Thus, the SRC framework in [23] was implemented using only raw pixels as features. However, under unconstrained variations and large distortions, the raw pixels are not informative enough to recover the sparse representation. In this work, it is shown that features that are more resilient to scene-dependent variations, [65, 104, 8, 101, 38, 102, 105], help in keeping the framework classes distinct, and can result in a significant performance gain under quality distortions.

Wright *et al.* [23] considered in their work the effect of random pixel corruption and occlusion by modifying the model described in (2.20) to explicitly account for additive noise w:

$$y = D.\alpha^* + w \tag{3.1}$$

where $w \, \epsilon \, \mathbb{R}^{d \times 1}$. A fraction ρ of w's entries are non-zero corresponding to the corrupted or occluded pixels in y. (3.1) can be rewritten as [23]:

$$y = \begin{bmatrix} D & I \end{bmatrix} \begin{bmatrix} \alpha^* \\ w \end{bmatrix} = A.w_0 \tag{3.2}$$

where $A = [D \;\; I]$ and $w_0 = [\alpha^* \; w]^T$. w_0 has at most $n_i + \rho d$ non-zero entries, where n_i is the number of non-zero entries of α^* for the i^{th} class and ρd is the number of non-zero entries of w. As stated in [23], we might recover w_0^* as the sparsest solution for (3.2) by solving the l_1-norm minimization problem (2.21) as long as the fraction of occlusion is less than 33 percent of the image size d (i.e., $\rho < 33\%$). This implies that if the occlusion covers a large portion of the image (more than 33 percent coverage), the SRC framework cannot provide the sparsest solution for (3.2), and thus, its performance decreases.

3.3 Augmented SRC (ASRC)

The different distortion types that are considered in this work are first described and their effect on sparse representations is derived. A modification to SRC is then proposed, namely ASRC, to further enhance the classification performance under these types of distortion.

3.3.1 Effect of Distortions on Sparse Representations

Let $y \in \mathbb{R}^{d \times 1}$ be a raw pixel test sample that can be sparsely represented in terms of the training samples in $D \in \mathbb{R}^{d \times K}$ as given in (2.20). If $D_i \in \mathbb{R}^{d \times 1}$ is the i^{th} atom entry vector of D and x_i is the i^{th} entry element of $\alpha^* \in \mathbb{R}^{K \times 1}$, (2.20) can then be rewritten as:

$$y = x_1 D_1 + x_2 D_2 + ... + x_K D_K \tag{3.3}$$

Let $\mathcal{F}(.)$ be an operation representing the distortion applied to y and whose output is $\widehat{y}$, the distorted version of y, as follows:

$$\widehat{y} = \mathcal{F}(y) = \mathcal{F}(x_1 D_1 + x_2 D_2 + ... + x_K D_K) \tag{3.4}$$

If the operation $\mathcal{F}(.)$ is linear, $\widehat{y}$ can be expressed as:

$$\widehat{y} = x_1\widehat{D}_1 + x_2\widehat{D}_2 + ... + x_K\widehat{D}_K \tag{3.5}$$

Let $\widehat{D} \in \mathbb{R}^{d\times K}$ be the dictionary whose atoms vectors are $\widehat{D}_i = \mathcal{F}(D_i), 1 \leq i \leq K$. From (3.3) and (3.5), it follows that if y is sparsely represented in terms of the elements in D, the distorted version $\hat{y} = \mathcal{F}(y)$ will be sparsely represented in terms of the elements in $\widehat{D}$. Furthermore, y and $\widehat{y}$ will have the same sparse coefficient vector α^*.

To illustrate the distortions that can be described by a linear operation, the cases of blur and block occlusion are considered. The case of non-sparse additive noise is also described in addition of how this case can be handled.

3.3.1.1 Effect of Blur

The blurring process can be represented using a point spread function (PSF) where a special case is a Gaussian distribution function, which is written as:

$$h(u,v) = \frac{1}{2\pi\sigma^2}e^{-}\frac{u^2+v^2}{2\pi\sigma^2} \tag{3.6}$$

where σ is the standard deviation and (u,v) denotes the pixel location at which the blur PSF is defined. The Gaussian blur PSF accounts for blur due to mainly atmospheric turbulence, such as fog or rain. For convenience, it will be referred to the lexicographically ordered $h(u,v)$ as h, where the blur level is controlled by the variance σ^2, and to $*$ as a 2D convolution followed by a lexicographic ordering. The blur distortion can thus be represented using the linear operation $\mathcal{F}(y) = y * h$.

Consequently, using (3.5), the blurred image $\widehat{y}$ can be expressed as:

$$\widehat{y} = x_1(D_1 * h) + x_2(D_2 * h) + ... + x_K(D_K * h) \tag{3.7}$$

The SRC framework described in Section 2.3.2 shows that a test sample y can be sufficiently represented using only the training samples from the same class as in (2.20) if the representation is naturally sparse. The more sparse the recovered α^* is, the easier will it be to accurately determine the identity of the test sample y. Under the condition that the considered blur level preserves sufficient separation between the M classes to keep these distinct, (3.7) indicates that if the clean image y is sparsely represented in terms of the clean training samples in the dictionary D, then blurring the atoms in D at the same blur level σ as the blurred test sample $\widehat{y}$ will maintain the sparsity of the representation.

3.3.1.2 Effect of Block Occlusions

In many practical face recognition scenarios, the test image y could be partially occluded by one or several blocks due to the loss of data packets during the transmission of compressed images over lossy packet networks. Face recognition is particularly affected if the blocks are obstructing major facial features. In [23], the SRC framework is designed to uniformly handle small-size sparse occlusions by considering such occlusions with an additive operation as in (3.1). In this work, to investigate the resilience of the SRC framework to less sparse occlusions (covering more than 33 percent of the image size), these are represented as a linear pointwise vector multiplication operation. Let $y \, \epsilon \, \mathbb{R}^{d \times 1}$ be a raw-pixel test sample that can be sparsely represented in terms of the training samples in $D \, \epsilon \, \mathbb{R}^{d \times K}$ as given in (3.3). A single block occlusion of size k is modeled by considering an occlusion kernel $b(u, v)$, where $b(u, v) = c(u, v) \neq 1$ for $u_1 \leq u \leq u_2$, $v_1 \leq v \leq v_2$, $(u_2 - u_1)(v_2 - v_1) = k$, and $b(u, v) = 1$ elsewhere. (u, v) denotes the pixel location at which the kernel is defined. For example, a black occlusion occurs if $c(u, v) = 0$. $b(u, v)$ can represent multiple block occlusions as the sum of two or more single block occlusions. For convenience, it will be referred to

the lexicographically ordered $b(u, v)$ as b and to $\times$ as a 2D pointwise multiplication followed by a lexicographic ordering. As such, the block occlusion distortion can be represented by a linear operation $\mathcal{F}(y) = y \times b$.

From (3.5), it follows that the occluded image $\widehat{y}$ can be expressed as:

$$\widehat{y} = x_1(D_1 \times b) + x_2(D_2 \times b) + ... + x_K(D_K \times b) \tag{3.8}$$

Under the assumption that the considered occlusion preserves sufficient separation between the M classes to keep these distinct, (3.8) indicates that if the clean image y is sparsely represented in terms of the clean training samples in the dictionary D, then adding occlusion to the atoms in D at the same block position and with the same size k as the occluded test sample $\widehat{y}$ will maintain the sparsity of the representation.

It is worth mentioning that representing the atoms using features that are more resilient to blur or occlusion, as compared to raw pixels, will help in keeping the classes distinct at higher blur/occlusion levels.

3.3.1.3 Effect of Additive Noise

In many scenarios, the corrupted image $\widehat{y}$ results from a pointwise additive operation $\widehat{y} = \mathcal{F}(y) = y + w$. A distortion resulting from a Gaussian white noise is an example of an additive operation where w is a collection of independent identically distributed real-valued random variables following a Gaussian distribution with mean $m = 0$ and variance σ^2.

An additive distortion is handled well by the SRC, as long as the corruption coverage of w does not exceed 33 percent of the image size, which is required to preserve the sparsity of the SRC representation as explained previously in Section 3.2. The way to handle larger non-sparse additive noise will be discussed in Section 3.3.2.

3.3.2 Proposed Method

In the following, the Augmented SRC method (ASRC) is presented in details. The proposed method aims at improving the performance of the original SRC [23] under blur, block occlusion, and non-sparse additive noise.

3.3.2.1 ASRC under Blur and Occlusion

In this section, the ASRC model is proposed for the blur and block occlusion distortions. The discussion here applies as well to other types of distortions that can be described or approximated using linear operations. From (3.7) and (3.8), it can be seen that the distorted raw image test sample $\widehat{y}$ can be sparsely represented in terms of the distorted atoms $\mathcal{F}(D_i), 1 \leq i \leq K$, if the clean test sample y is sparsely represented in terms of the atoms $D_i, 1 \leq i \leq K$, and if the classes are kept distinct after applying the operation $\mathcal{F}(.)$ to the dictionary atoms. This brings the importance of representing the atoms with features that are more resilient to blur/occlusion distortion as compared to raw pixels, as discussed in more details later in this work.

For the blur distortion, in order to accommodate various blur levels, it is proposed to augment the dictionary D with training samples that have been blurred with N_d Gaussian blur kernels of increasing variance σ^2 and size (including absence of blur). For the occlusion distortion, and in order to accommodate various occlusion positions, it is proposed to augment the dictionary D with training samples, which have been occluded at N_d different occlusion positions with a specified occlusion size (including absence of occlusion). In either case, it is opted to assign the group of atoms that corresponds to a specific distortion level (blur) or distortion position (occlusion), for a particular identity i, $1 \leq i \leq M$, to a separate object class j, $1 \leq j \leq N_d M$, in the resulting dictionary that will be denoted by $\widehat{D}$. For simplicity, in the remainder of this

section, both the blur level and the occlusion position will be denoted by distortion level $l, 0 \leq l \leq N_d - 1$, for short.

It is assumed that the original dictionary D consisting of the clean training samples is given by:

$$D = \left[\bar{D}_1 \, |\bar{D}_2 \, |... \, |\bar{D}_M\right] \tag{3.9}$$

where $\bar{D}_i = \{D_1^i, D_2^i, \cdots, D_{m_i}^i\}$, $1 \leq i \leq M$, is the set of training samples for the i^{th} object class. Let $\mathcal{F}_l(.)$ be a linear operation representing the distortion at level $l, 0 \leq l \leq N_d - 1$. Also let $\mathcal{F}_l(\bar{D}_i) = \{\mathcal{F}_l(D_1^i), \mathcal{F}_l(D_2^i), \cdots, \mathcal{F}_l(D_{m_i}^i)\}$. The proposed dictionary $\widehat{D}$ is given by:

$$\widehat{D} = \left[\widehat{D}_1 \, |\widehat{D}_2 \, |... \, |\widehat{D}_{N_d M}\right] \tag{3.10}$$

where N_d corresponds to the number of represented distortion levels and $\widehat{D}_j$, $1 \leq j \leq N_d M$, is given by:

$$\widehat{D}_j = [\mathcal{F}_{l=(j-1) mod N_d}(\bar{D}_{(j-1)\backslash N_d + 1})] \tag{3.11}$$

In (3.11), $\backslash$ and mod correspond to the integer division and modulo operations, respectively, and $l = 0$ corresponds to the absence of distortion. The SRC is applied using the proposed dictionary to classify test samples based on the class-wise minimum reconstruction error as in (2.24). Nevertheless, as the subject belongs to either one of the N_d possible levels (including absence) of distortion, the subject identity i is obtained as follows:

$$i = ((j^* - 1)\backslash N_d) + 1 \tag{3.12}$$

where j^* is computed using (2.24), except that the residual r_j is now computed with respect to the new constructed dictionary $\widehat{D}$ and M is replaced by $N_d M$. Algorithm 1 summarizes the procedure of the ASRC method for the blur and occlusion distortions.

Algorithm 1: The ASRC Algorithm for Blur and Occlusion

Input: Training samples $\mathbf{D} = |\bar{\mathbf{D}}_1\ \bar{\mathbf{D}}_2\ ...\ \bar{\mathbf{D}}_M|$ for M subjects and test sample $\widehat{\mathbf{y}}$, where $\bar{D}_i = \{D_1^i, D_2^i, \cdots, D_{m_i}^i\}$, $1 \leq i \leq M$, is the set of training samples for the i^{th} object class.

Step1: Distort (blur and/or occlude) the training samples of $\mathbf{D}$ by applying N_d levels of distortions represented by linear operations $\mathcal{F}_l(.), 0 \leq l \leq N_d - 1$, to each training sample in D:

for $j = 1 : N_d M$ **do**

  $\widehat{\mathbf{D}}_j = \mathcal{F}_{(j-1)modN_d}(\bar{\mathbf{D}}_{(j-1)\backslash N_d+1})$;

end

Step 2: Generate the distortion-augmented dictionary $\widehat{\mathbf{D}}$:

$\widehat{\mathbf{D}} = [\widehat{\mathbf{D}}_1\ \widehat{\mathbf{D}}_2\ ...\ \widehat{\mathbf{D}}_{N_d M}]$

Step 3: Use SRC to classify $\widehat{\mathbf{y}}$:

$\alpha^* = arg\ \min_\alpha\ ||\widehat{\mathbf{y}} - \widehat{\mathbf{D}}\alpha||_2^2\ + \lambda||\alpha||_1$

for $j = 1 : N_d M$ **do**

  $r_j = \min_j(||\widehat{\mathbf{y}} - \widehat{\mathbf{D}}\delta_j(\alpha^*)||_2^2)$

end

$j^* = arg\ \min_j\ (r_j)$

Output: identity($\mathbf{y}$) $\leftarrow$ $(j^* - 1)\backslash N_d + 1$

3.3.2.2 ASRC under Additive Noise

The proposed method is extended to solve the additive noise distortion $\mathcal{F}(y)$ applied to a clean test sample y, as described in Section 3.3.1.3 resulting in a noisy test $\widehat{y} = y + w$, where w is the additive noise. In this case, a lowpass filter h_{lp} of variance

Algorithm 2: The ASRC Algorithm for Additive Noise

Input: Training samples $\mathbf{D} = [\bar{\mathbf{D}}_1\ \bar{\mathbf{D}}_2\ ...\ \bar{\mathbf{D}}_M]$ for M subjects and test sample $\widehat{\mathbf{y}}$, where $\bar{D}_i = \{D_1^i, D_2^i, \cdots, D_{m_i}^i\}$, $1 \leq i \leq M$, is the set of training samples for the i^{th} object class.

Step 1: Denoise the test sample $\widehat{\mathbf{y}}$ by applying a lowpass filter $\mathbf{h}_{lp}$ with variance σ_n^2:

$\tilde{\mathbf{y}} = \widehat{\mathbf{y}} * \mathbf{h}_{lp}$

Step 2: Perform Steps 1 and 2 of Algorithm 1 for the blur case

Step 3: Perform Step 3 of Algorithm 1 where $\widehat{\mathbf{y}}$ is replaced by the denoised test sample $\tilde{\mathbf{y}}$.

Output: identity$(\mathbf{y}) \leftarrow (j^* - 1) \backslash N_d + 1$

σ_n^2 is first applied to the noisy test sample in order to reduce the noise:

$$\tilde{y} = \widehat{y} * h_{lp} \tag{3.13}$$

The variance of the applied lowpass filter h_{lp} is computed using a noise level estimation method, such as [106, 107, 108]. $\tilde{y}$ is a denoised blurred version of the original test image y. In this way, the non-sparse additive noise problem can be transformed into a blur problem, which can be effectively handled using the proposed ASRC method. The resulting test sample $\tilde{y}$ is sparsely represented in terms of the proposed dictionary $\widehat{D}$, as described previously in Section 3.3.2.1. Algorithm 2 summarizes the procedure of the ASRC method for the additive noise.

3.3.3 The Effect of Feature Extraction

The choice of feature extraction is essential to properly separate object classes in a more discriminative space, also known as feature space. Feature extraction methods are mainly subdivided into three main groups. The subspace learning group,

which includes raw image values, Eigenfaces [4], Fisherfaces [5], and Laplacianfaces [109], is a conventional class of linear features where training samples form a single class are modeled to lie on a linear subspace. This type of features has been widely used with classifiers, such as Nearest Neighbor (NN), Nearest Space (NS) and linear Support Vector Machines (SVM). The subspace learning methods have been replaced later by the hand-crafted non-linear approaches, which use the local orientation information. These features have proven to work well on face recognition applications in constrained environments. They include features, such as Local Binary Patterns (LBP) [104], Local Phase Quantisation (LPQ) [66], Histogram of Oriented Gradients (HoG) [65], and Fisher vectors [105]. More recently, DNN-based models have achieved high performance in terms of accuracy in the face recognition domain. They consist of layers of convolutional filters where the weights of the filters can be learned using a gradient descent-based optimization procedure. Deep features extracted from DNNs, such as DeepFace [101], DeepID [102], VGG-Face [8], and FaceNet [9], are discriminative enough to recognize clean face images that have been captured in unconstrained environments.

In [23], the authors designed and tested the SRC framework on subspace learning features for ease of presentation. Wright *et al.* [23] argue that, if the sparsity is well harnessed, the choice of the features is not important, as long as the dimension of the feature exceeds a specific bound [23].

In this work, a performance evaluation of the SRC and ASRC frameworks is provided by selecting different feature types. For this purpose, hand-crafted features (HoG), as well as deep learning features (VGG-Face) are considered for two main reasons. Hand-crafted features are effective at detecting local information, namely contour and texture, specifically in constrained environments. The deep learning features are robust in unconstrained environments, as the DNN layers have configurable

parameters that can be learned from the processed data across the layers. In Section 3.4, the experimental results verify that the choice of these two types of features affect the performances of the SRC and ASRC frameworks by improving their recognition accuracy rates compared to raw image features.

A feature transformation $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^m$ maps data from the image space of dimension d to a high-dimensional feature space of dimension m, where $d << m$. Consequently, SRC is performed by solving the following constrained l_1-norm minimization problem:

$$\alpha^* = arg \min_{\alpha} \quad ||\Phi(y) - \Phi(D).\alpha||_2^2 + \lambda||\alpha||_1 \tag{3.14}$$

where $\Phi(D) = [\Phi(\bar{D}_1)|\Phi(\bar{D}_2)|\cdots|\Phi(\bar{D}_M)]$ and D is defined as in (3.9). The feature-domain ASRC framework is modeled using (3.14) where D is replaced with $\widehat{D}$, such that $\Phi(\widehat{D}) = [\Phi(\widehat{D}_1)|\Phi(\widehat{D}_2)|\cdots|\Phi(\widehat{D}_{N_dM})]$, and where $\widehat{D}$ is defined as in (3.10).

3.3.4 Feature Selection and Proposed Feature Sparse Coding and Classification Index (FSCCI)

In order to assess the quality of features in improving the sparsity of the representation, a method of feature selection is proposed based on the sparsity and the fidelity of the proposed ASRC framework.

The validity of a sample depends on the sparsity of its coefficient vector α^*, which can be measured using the sparsity concentration index (SCI). A valid test image has a sparse representation with nonzero entries concentrated mostly on one subject, whereas an invalid image has a sparse representation with coefficients spread among a wide range of multiple subjects. The SCI index of a coefficient vector α is defined as a measure of how concentrated the coefficients are on a single class in the dataset

and is computed as follows:

$$SCI(\alpha) = \frac{M \max_i ||\delta_i(\alpha)||_1 / ||\alpha||_1 - 1}{M - 1} \tag{3.15}$$

where M is the total number of distinct classes.

For a solution α^* as found in (2.21), $SCI(\alpha^*)$ varies between 0 and 1, where higher values correspond to sparser representations. As in [23], a threshold $\tau \, \epsilon \, (0, 1)$ is chosen and a sparse representation is accepted as valid if $SCI(\alpha^*) \geq \tau$.

The most discriminative feature should not only ameliorate the sparsity of the representation, but it must also improve the representation accuracy. The residual r is used to measure the fidelity of the representation by measuring the similarity between the test sample and each individual class.

Given an application with a validation dataset, it is aimed to rank the features based on their ability to increase the sparsity of the proposed framework representation while improving its identification performance. For this purpose, a feature sparse coding and classification index (FSCCI) is proposed for feature selection.

It is proposed to capture the number of valid images that are correctly classified (True Positive Rate or TPR) and the number of valid images that are incorrectly classified (False Positive Rate or FPR) for each threshold $\tau \, \epsilon \, (0, 1)$. For a given distorted validation data set $Y \epsilon \, \mathbb{R}^{d \times N}$ with N total samples, the following labels $\{z_1, z_2, ..., z_i, ..., z_N\}$ are assigned to each sample $y_i \, \epsilon \, Y$ and each τ value, where $z_i = \{-1, 0, 1\}$. All valid and correctly classified samples are labeled 1, all invalid samples are labeled 0, and all valid and incorrectly classified samples are labeled -1.

The number of true positives (TP) is computed as the total number of samples labeled 1 and the number of false positives (FP) as the total number of samples labeled -1. The TPR and FPR are subsequently calculated as follows:

$$TPR = \frac{TP}{TP + FN} \tag{3.16}$$

$$FPR = \frac{FP}{FP + TN} \tag{3.17}$$

In (3.16) and (3.17), FN and TN are, respectively, the numbers of false negatives and true negatives and their sum is the total number of samples labeled 0. While FN corresponds to the number of invalid and correctly classified samples, TN corresponds to the number of invalid and incorrectly classified samples.

For each feature, the area under the receiver operating-characteristic (ROC) curve (AUC) is computed as the FSCCI to capture the feature discriminatory ability. Therefore, FSCCI values vary from 0 (lowest feature rank) to 1 (highest feature rank).

3.4 Experimental Setup and Results

3.4.1 Datasets

The experiments are performed on three publicly available face recognition datasets: ORL [33], Extended Yale B [2], and LFW [3]. The first two datasets consist of images that are captured in a controlled environment where the image parameters that are allowed to change are limited to expression, pose, illumination and simple disguise, such as eyeglasses. The third dataset is more challenging, as it includes images that are captured in the "wild" under realistic unconstrained conditions.

3.4.1.1 ORL Dataset

The dataset includes 400 face images taken from 40 subjects (10 face images for each subject) [33]. For some subjects, the images were taken at different times, with

varying lighting, facial expressions, and facial details. All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position with tolerance for some side movement (Fig. 3.1). The size of each image is 92x112 pixels, with 256 gray levels per pixel.

3.4.1.2 Extended Yale B Dataset

The dataset contains 2,432 front face images of 38 individuals and each subject having around 64 near frontal images under different illuminations [2]. The main challenge of this dataset is to overcome extreme varying illumination conditions that were laboratory-controlled [110] (Fig. 3.2). The facial portion of each original image was cropped to a 192×168 image by the original authors.

3.4.1.3 LFW Dataset

The dataset [3] contains 13,233 images of 5,749 people captured and designed for unconstrained face recognition with dramatic variations of pose, illumination, expression, misalignment, and occlusion (Fig. 3.3(a)). The faces are collected from the web and detected by the Viola-Jones face detector.

3.4.2 Experimental Protocols

The procedure to produce the training and test datasets of the three considered face benchmarks is described next.

To evaluate the performance of the proposed method on the ORL dataset, half of the images is randomly selected from each class for training and the remaining for testing.

For the Extended Yale B, the face images are resized to 96×84 to reduce the computational cost. The dataset is randomly splitted into two halves. One half (32

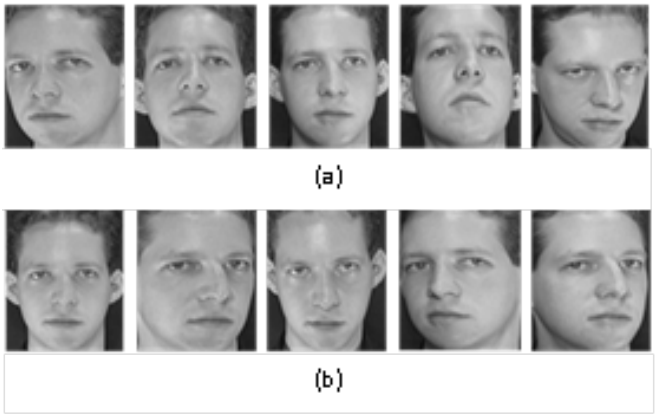


Figure 3.1: Example images randomly selected from the ORL dataset for a subject. (a) Training images. (b) Test images.

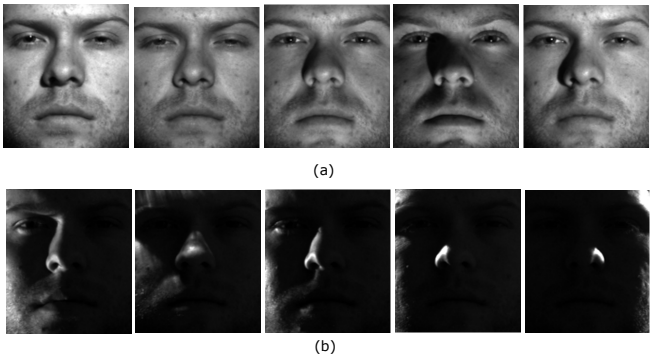


Figure 3.2: Example images randomly selected from the Extended Yale B dataset for a subject. (a) Training images. (b) Test images.

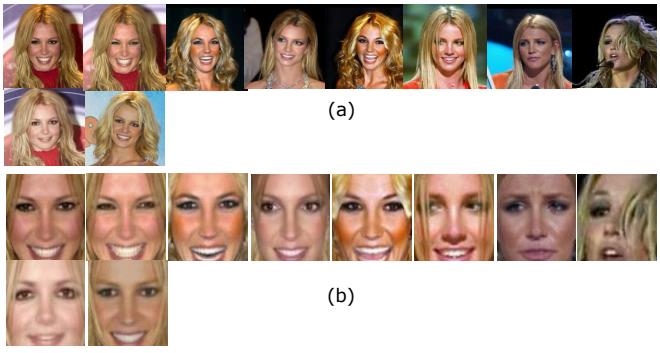


Figure 3.3: Example training images randomly selected from the identification LFW dataset for a subject. (a) Original LFW images. (b) Cropped and frontalized LFW images [10].

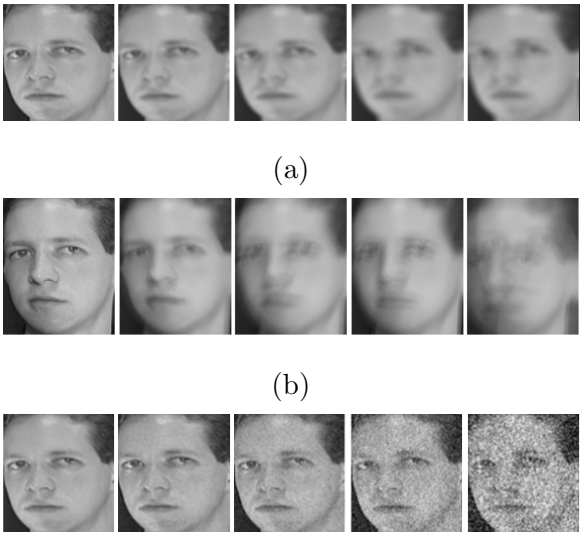
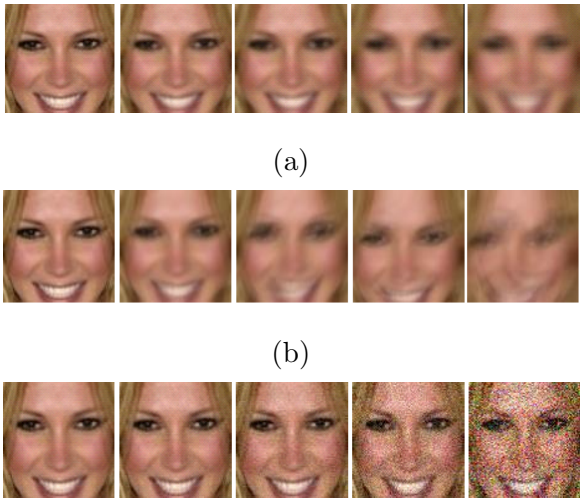


Figure 3.4: Corrupted images selected from the ORL dataset for a subject. (a) From left to right. Original image and its corresponding Gaussian blurred corrupted images at levels 1 to 4. (b) From left to right. Original image and its corresponding camera shake blurred corrupted images at 4 camera shake blur levels. (c) From left to right. Original image and its corresponding White noise corrupted images at levels 1 to 4.

Figure 3.5: Corrupted images selected from the Extended Yale B database for a subject. (a) From left to right. Original image and its corresponding Gaussian blurred corrupted images at levels 1 to 4. (b) From left to right. Original image and its corresponding camera shake blurred corrupted images at 4 camera shake blur levels. (c) From left to right. Original image and its corresponding White noise corrupted images at levels 1 to 4.

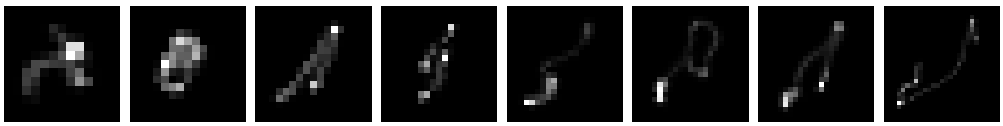


(a)

(b)

(c)

Figure 3.6: Corrupted images selected from the LFW face identification dataset for a subject. (a) From left to right. Original image and its corresponding Gaussian blurred corrupted images at levels 1 to 4. (b) From left to right. Original image and its corresponding camera shake blurred corrupted images at 4 camera shake blur levels. (c) From left to right. Original image and its corresponding White noise corrupted images at levels 1 to 4.

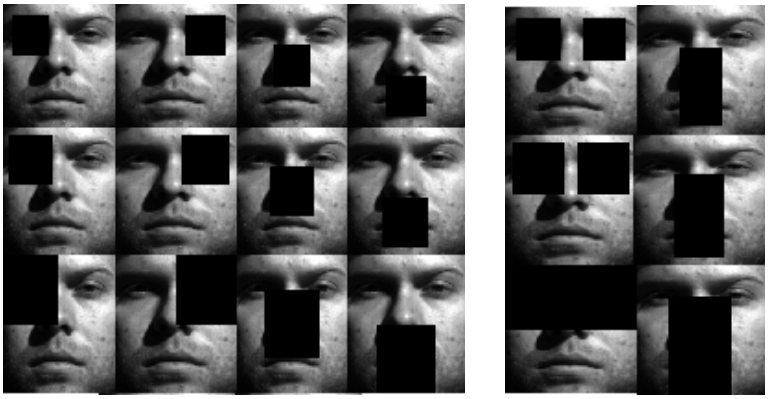


(a) (b) (c) (d) (e) (f) (g) (h)

Figure 3.7: The 8 blur kernels extracted from Levin *et al.* [1]. They are used to simulate realistic blur resulting from camera shake at eight distortion levels. Kernel sizes from left to right. (a) 13×13. (b) 15×15. (c) 17×17. (d) 19×19. (e) 21×21. (f) 23×23. (g) 23×23. (h) 27×27.

images from each class) for training and the other half for testing.

The original LFW dataset, which was constructed for face matching rather than identification, is rearranged to generate a face identification dataset, which consists of 5,088 images corresponding to 255 subjects. 70% samples per subject are randomly selected for training, and the rest for testing. The images are cropped and frontalized using the method proposed in Hassner *et al.* [10] (Fig. 3.3(b)). Then, the images are converted to grayscale, and resized to half their size from 90×90 to 45×45. The



(a) (b)

Figure 3.8: Occluded images selected from the Extended Yale B dataset for a subject. (a) From top to bottom single occluded images where each occlusion block size is 10%, 15%, and 25% of the image size, respectively. (b) From top to bottom double occluded images where each total occlusion size is 30%, 40%, and 50% of the image size, respectively.

constructed LFW face identification dataset will be made available at ivulab.asu.edu.

The proposed ASRC is tested on two main feature descriptors in addition to image raw pixels: HoG (hand-crafted) and VGG-Face (deep learning). HoG features have been successfully applied to face recognition [111, 112] for their geometric and photometric invariance property. HoG features can be found in two main variants, the Dalal *et al.* [65] original variant and the slightly improved Felzenszwalb *et al.* [69] UIUC variant. In this work, the UIUC HoG formulation [69] is used, which computes both the directed and undirected local gradients as well as a four dimensional texture-energy feature for each cell, resulting in a larger HoG feature's dimension as compared to [65]. The histogram of orientations includes 9 bins for the undirected orientations and 27 for the directed ones. The final dimension of the HoG feature for one block is projected down to 31 dimensions as described in [69]. This variant accounts more than the original one for rotations and translations by considering both directed and undirected orientations. It also includes the texture-energy feature, which improves the performance in capturing local information in the face images. In this work,

for an image consisting of N blocks, N HoG feature vectors are computed, each of dimension 31. The final HoG descriptor is generated from the lexicographic ordering of the N HoG feature vectors. In our implementation, a block size of 8×8 was used as in [69].

VGG-Face deep features are also extracted to evaluate the performance of the proposed method. The VGG-Face CNN architecture A described in [8], which is trained on a large diverse dataset of face images, is particularly adopted to transfer its extracted features to the face recognition problem. The deep features are extracted from the last convolutional layer of the 16-layer deep network. As the neural network operates on a $224 \times 224 \times 3$ size input, the original images are resized to this dimension before feeding them to the network. The VGG-Face mean image is additionally subtracted from the input images of the image dataset that was used in [8] to train the VGG-Face network.

To evaluate the performance, three metrics are computed: the recognition rate accuracy in addition to the mean SCI and the proposed FSCCI, as described in Section 3.3.4. While the recognition rate evaluates how well the representation approximates the test sample by relying on the residuals, the SCI evaluates how good the representation itself is based on the localization of the sparse coefficients. The proposed FSCCI combines both previous metrics by measuring the representation sparsity and the reconstruction fidelity.

3.4.3 Addition of Distortions to Datasets

A Gaussian blur function is simulated as in (3.7) and convolved with the face images. The filter size of the Gaussian filter is set in number of pixels as [5, 5, 7, 9] for the different blur levels represented by the blur variance values [1, 2, 4, 8],

respectively. The levels of distortions are carefully chosen to generate images covering a broad range of quality, from imperceptible levels to high levels of impairment. The dictionary is augmented with the same four Gaussian blur levels in addition to the original clean images.

Next the robustness of the ASRC is tested to unseen distortions by considering camera shake blur, which is more general than the previously considered Gaussian blur, because the blur kernel is not symmetric. The 8 blur kernels provided by [1], which were captured from a real camera, are used and convolved with the test images of the three considered datasets ((Figs. 3.4(b), 3.5(b) and 3.6(b)). The blur kernels have different shapes and different sizes including 13×13, 15×15, 17×17, 19×19, 21×21, 23×23 and 27×27, as shown in Fig. 3.7. The kernels in Figs. 3.7(f) and 3.7(g) have the same size but different shape. These 8 blur kernels result from the relative motion of a camera mounted on a tripod (z-axis) with loosened x and y handles. The motion is an in-plane rotation (rotation around the z-axis), which is a significant component of human hand shake. For the camera shake distortions, the robustness of the ASRC framework is tested to unseen distortions by using the Gaussian blur ASRC (GB-ASRC) dictionary structure. The GB-ASRC augments the ASRC dictionary with Gaussian blurred images at the same four distortion levels as described before. Then, a camera shake blur ASRC (CSB-ASRC) dictionary structure is considered. The CSB-ASRC augments the ASRC dictionary with realistic camera shake blurred images resulting from four of the 8 blur kernels of Levin *et al.* [1]. The sizes of the four selected blur kernels are 13×13, 17×17, 21×21 and 27×27.

Several block occlusion sizes are simulated ranging from 10 percent to 50 percent, by replacing one or two blocks in each test image with one or two black boxes at major facial locations, including the eyes, the nose and the mouth, as in Fig. 3.8. Therefore, the considered occlusions are either single (Fig. 3.8(a)) or double (Fig. 3.8(b)).

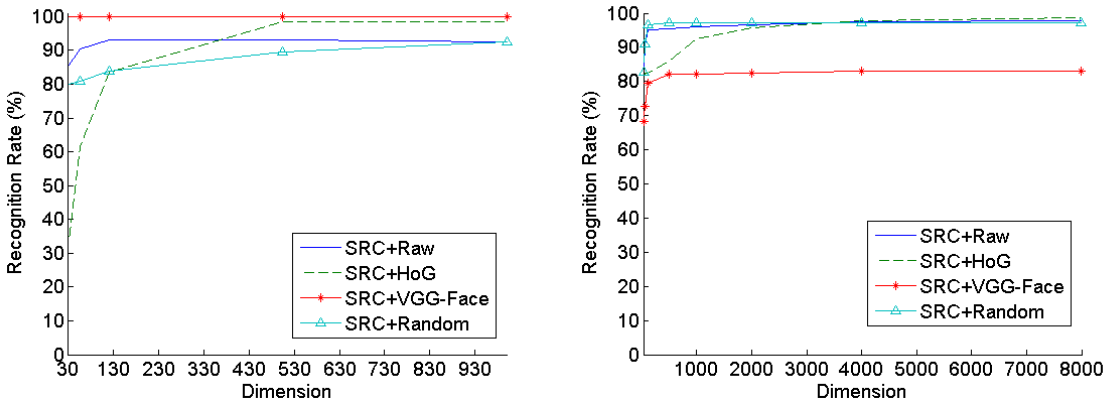
Moreover, single contiguous blocks are also randomly added to the face images. The dictionary in all the considered cases is augmented, in addition to the original images, with four single-block occluded images at four specified positions including the left eye, the right eye, the nose and the mouth. To test the robustness of the ASRC to unseen occlusions, the size of the block occlusion in the ASRC dictionary is fixed to 10 percent of the image size.

Finally, the additive Gaussian noise is simulated with zero mean and variance σ^2 (Section 3.3.1.3). Four levels of white Gaussian noise are added to the face images, where the variance values are [5, 10, 20, 40] (Fig. 3.4(a) to Fig. 3.6(a)). Again, the levels of distortions are carefully chosen to generate images covering a broad range of quality, from imperceptible levels to high levels of impairment. The dictionary is augmented with the same previous four Gaussian blur levels in addition to the original clean images.

3.4.4 Results

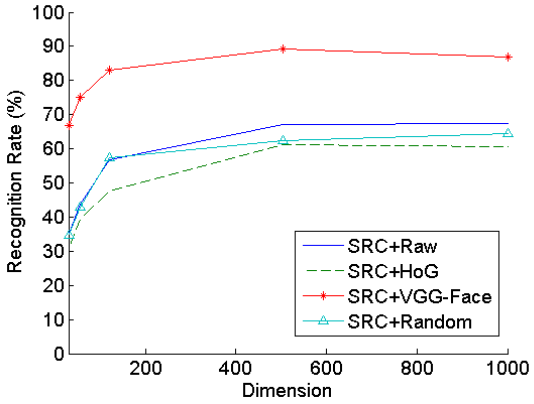
3.4.4.1 Feature Selection and its Impact on Sparse Representation

The feature choice and its impact on the performance of SRC is evaluated for the feature space dimensions 30, 56, 120, 504, 1,000, 2,000, 4,000 and 8,000. In [23], the authors use linear feature transformations, such as Eigenfaces, Laplacianfaces and Fisherfaces, in addition to raw image pixels. For the raw image pixels, the feature space dimension is reduced by downsampling the images appropriately. In this work, the feature dimension size is reduced for all the considered feature types using Principal Component Analysis (PCA), while keeping the same initial image size. Fig. 3.9 shows the SRC recognition performance for the various features (Raw, HoG, VGG-Face) in addition to the randomly sampled faces (Randomfaces) [23], which are



(a) ORL

(b) Extended Yale B



(c) LFW

Figure 3.9: Recognition rate (%) of raw, Randomfaces, HoG and VGG-Face features for different dimension sizes. (a) ORL (b) Extended Yale B (c) LFW. Different colors and symbols represent the different feature types.

considered as a less-structured counterpart to classical face features.

In the case of the ORL dataset, it is observed that, while the recognition accuracy increases for the raw, HoG and Randomfaces features and becomes stable for a dimension size larger than 530, the VGG-Face demonstrates a steady performance of 100% for all the considered dimension sizes. It is also worth noting that the HoG feature outperforms the raw and Randomfaces starting from a dimension size of 400. Applying (2.25) with $t = 5$ (number of training samples per class) and $K = 200$ (total number of training images) shows that a min dimension of d of 14 is required to

have a sparse solution that can be correctly recovered via l_1-minimization regardless of the feature type. However, the results show that the performance of the different feature types becomes steady for a dimension larger than 500. Moreover, it is worth noting that the recognition accuracies of VGG-Face and HoG features are better than the raw and RandomFaces features and that they do not depend on the computed dimension size of (2.25).

For the Extended Yale B dataset, the raw, RandomFaces and HoG features show a steady similar performance for a dimension size of 2,000. However, the VGG-Face feature shows a different trend with a recognition accuracy that is lower than the other features for all dimension sizes and that becomes stable starting from a dimension size of 503. Applying (2.25) with $t = 38$ (number of training samples per class) and $K = 1,181$ (total number of training images) shows that the required dimension size to have a sparse enough solution is 90.

Finally, for the LFW dataset, while the raw, Randomfaces and HoG features show a similar performance starting from a dimension size of 503, the VGG-Face feature outperforms the other features at all dimension sizes. Computing the min required dimension size using (2.25) with $t = 14$ (average number of training samples per class) and $K = 3,577$ (total number of training images) gives a value of 52.

In the next series of experiments, the effect of feature choice is evaluated on the performance of the ASRC framework in the presence of different levels of Gaussian blur. For these experiments, the representation ability of each feature (raw, HoG and VGG-Face) is analyzed with respect to the blur level. To quantify the observations, the mean SCI is first computed, which measures the average sparsity level of the representations for every blur level. The proposed feature sparse coding and classification index (FSCCI) is also evaluated, which is compared to the mean SCI. Finally, the results are validated by displaying the accuracy values for the three considered

Table 3.2: Mean SCI, FSCCI and Recognition Accuracy (%) for the ASRC framework Using Raw, HoG and VGG-Face Features on ORL, Extended Yale B and LFW Datasets. The Gaussian Blur Level in the Test Images Varies from an Imperceptible Level (1) to a High Impairment Level (4). Bold Entries Show the Highest Values for the Mean SCI, the FSCCI and the Recognition Accuracy for Each Blur Level.

		Mean SCI				**FSCCI**				**Accuracy (%)**			
		Blur level				Blur level				Blur level			
Dataset	**Method**	level 1	level 2	level 3	level 4	level 1	level 2	level 3	level 4	level 1	level 2	level 3	level 4
ORL	ASRC+raw	0.5706	0.5686	0.5562	0.5358	0.9263	0.9284	0.9338	0.9371	92.50	93.00	93.50	93.50
	ASRC+HoG	0.6109	0.6313	0.6439	0.6262	0.9905	0.9948	0.9792	0.9436	97.00	96.50	96.00	97.50
	ASRC+VGG-Face	**0.7740**	**0.8001**	**0.7938**	**0.7236**	**0.9975**	**0.9975**	**0.9975**	**0.9975**	**100.0**	**100.0**	**100.0**	**100.0**
Extended Yale B	ASRC+raw	**0.4707**	**0.4632**	**0.4430**	**0.4164**	0.9769	0.9736	0.9616	0.9508	98.71	98.23	97.35	95.82
	ASRC+HoG	0.4044	0.4096	0.4123	0.4009	**0.9784**	**0.9807**	**0.9747**	**0.9747**	**98.79**	**98.71**	**98.23**	**97.59**
	ASRC+VGG-Face	0.4582	0.4537	0.4356	0.3943	0.9543	0.9508	0.9416	0.9393	81.01	80.93	78.60	74.09
LFW	ASRC+raw	0.1836	0.1828	0.1744	0.1633	0.8405	0.8393	0.8479	0.8563	45.40	44.41	41.76	38.19
	ASRC+HoG	0.1537	0.1550	0.1561	0.1494	0.8153	0.8118	0.8301	0.8353	38.12	38.39	36.14	35.21
	ASRC+VGG-Face	**0.3445**	**0.3595**	**0.2983**	**0.2136**	**0.9211**	**0.9209**	**0.9231**	**0.8890**	**81.01**	**83.39**	**71.48**	**59.63**

datasets. The cases where the test samples have been blurred at four different Gaussian blur levels, varying from an imperceptible level (1) to a highly impaired level (4) are considered. The corresponding results are listed in Table 3.2.

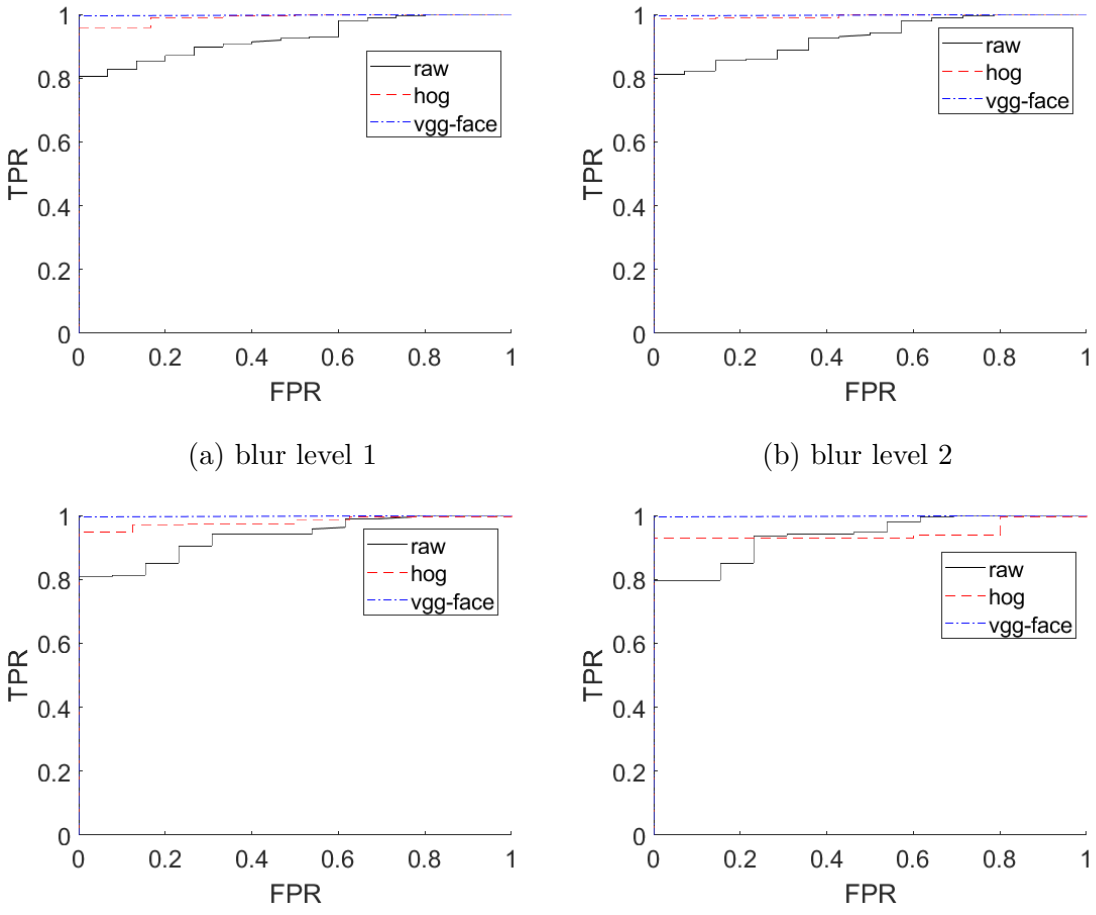
For the ORL dataset, the ASRC achieves the best recognition rates with VGG-Face at different blur levels followed by HoG and raw images. This is reflected by the mean SCI and FSCCI values, which are the highest with VGG-Face. For the Extended Yale B dataset, it is observed that, at all four blur levels, HoG's recognition accuracy is the highest as compared to the raw images and the VGG-Face features. With the Extended Yale B dataset, which is impaired with extreme light variations, HoG proves its photometric invariance property. From Table 3.2, it can be seen that the VGG-Face representation is the most affected by these variations, as it is even outperformed by raw images. Furthermore, while the proposed FSCCI is able to correctly rank the features (highest for HoG and lowest for VGG-Face), the mean SCI is not a good indicator for the best feature to use. Finally, for the LFW dataset, the ASRC achieves the best recognition accuracy with the VGG-Face features at

Table 3.3: Mean SCI, FSCCI and Recognition Accuracy (%) for the ASRC Framework Using Raw, HoG and VGG-Face Features on ORL, Extended Yale B and LFW Datasets. The Occlusion Level in the Test Images Varies from 10 Percent to 25 Percent of the Image Size. Bold Entries Show the Highest Values for the Mean SCI, the FSCCI and the Recognition Accuracy for Each Occlusion Size.

		Mean SCI			**FSCCI**			**Accuracy (%)**		
		Occlusion size			Occlusion size			Occlusion size		
Dataset	**Method**	10%	15%	25%	10%	15%	25%	10%	15%	25%
ORL	ASRC+raw	0.3788	0.3462	0.2816	0.9360	0.9243	0.8443	85.50	83.37	71.00
	ASRC+HoG	0.5341	0.5002	0.4313	0.9646	0.9519	0.9560	96.37	96.00	94.37
	ASRC+VGG-Face	**0.7710**	**0.6987**	**0.5571**	**1.000**	**0.9979**	**0.9867**	**99.50**	**99.37**	**98.50**
Extended Yale B	ASRC+raw	0.3900	0.3575	0.2753	0.9655	0.9481	0.9035	96.68	94.97	86.49
	ASRC+HoG	**0.4206**	**0.3951**	**0.3180**	**0.9772**	**0.9691**	**0.9414**	**98.27**	**97.91**	**94.69**
	ASRC+VGG-Face	0.3613	0.3326	0.2768	0.9114	0.9091	0.8783	77.03	73.39	65.06
LFW	ASRC+raw	0.1483	0.1316	0.1129	0.8509	0.8451	0.8087	42.53	39.21	32.94
	ASRC+HoG	0.1543	0.1452	0.1449	0.8437	0.8234	0.7864	36.68	32.54	26.36
	ASRC+VGG-Face	**0.2483**	**0.2199**	**0.1709**	**0.8945**	**0.8743**	**0.8625**	**71.04**	**67.09**	**58.23**

all blur levels. This consolidates the robustness of the deep features when used for unconstrained image recognition. In all cases, the proposed FSCCI is able to correctly rank the features in terms of their discriminative ability and detection accuracy.

In the third series of experiments, the effect of the feature choice on the performance of the ASRC framework is evaluated in the presence of different levels of block occlusion. For these experiments, the representation ability of each feature is analyzed with respect to the occlusion level. Again, to quantify the observations, the mean SCI is first computed, which measures the average sparsity level of the representations for every occlusion size and is compared with the proposed FSCCI. Finally, the results are validated by displaying the accuracy values for the three considered datasets. The cases where the test samples have been occluded at four different positions including the left eye, the right eye, the nose and the mouth are considered. The considered occlusions sizes vary from 10 percent to 25 percent of the image size.



(a) blur level 1

(b) blur level 2

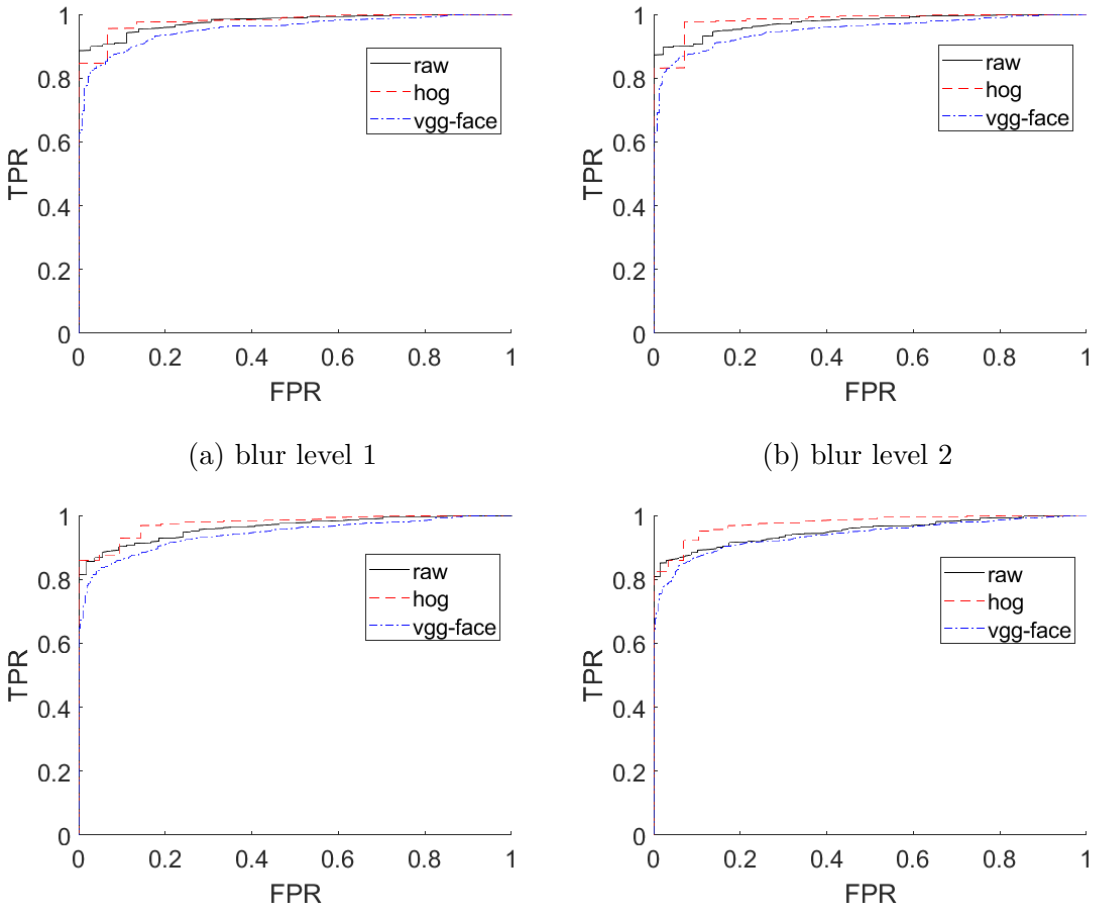
(c) blur level 3

(d) blur level 4

Figure 3.10: ROC curves of raw, HoG and VGG-Face for the ORL dataset. The test samples are blurred at (a) level 1 (b) level 2 (c) level 3 and (d) level 4. Different colors and symbols represent the different feature types. The ROC curves show that the VGG-Face feature better separates the classes by providing the highest AUC.

The corresponding results are listed in Table 3.3.

For the ORL dataset, both the mean SCI and the FSCCI values are the largest for the VGG-Face feature, which proves again that the VGG-Face feature is the most representative of the ORL dataset, followed by HoG, and then raw images. For the Extended Yale B, the mean SCI and FSCCI values show that HoG performs the best while VGG-Face performs the least. Again, this confirms the previous results for the blur distortion. Finally, for the LFW dataset, the mean SCI and the FSCCI values



(a) blur level 1

(b) blur level 2

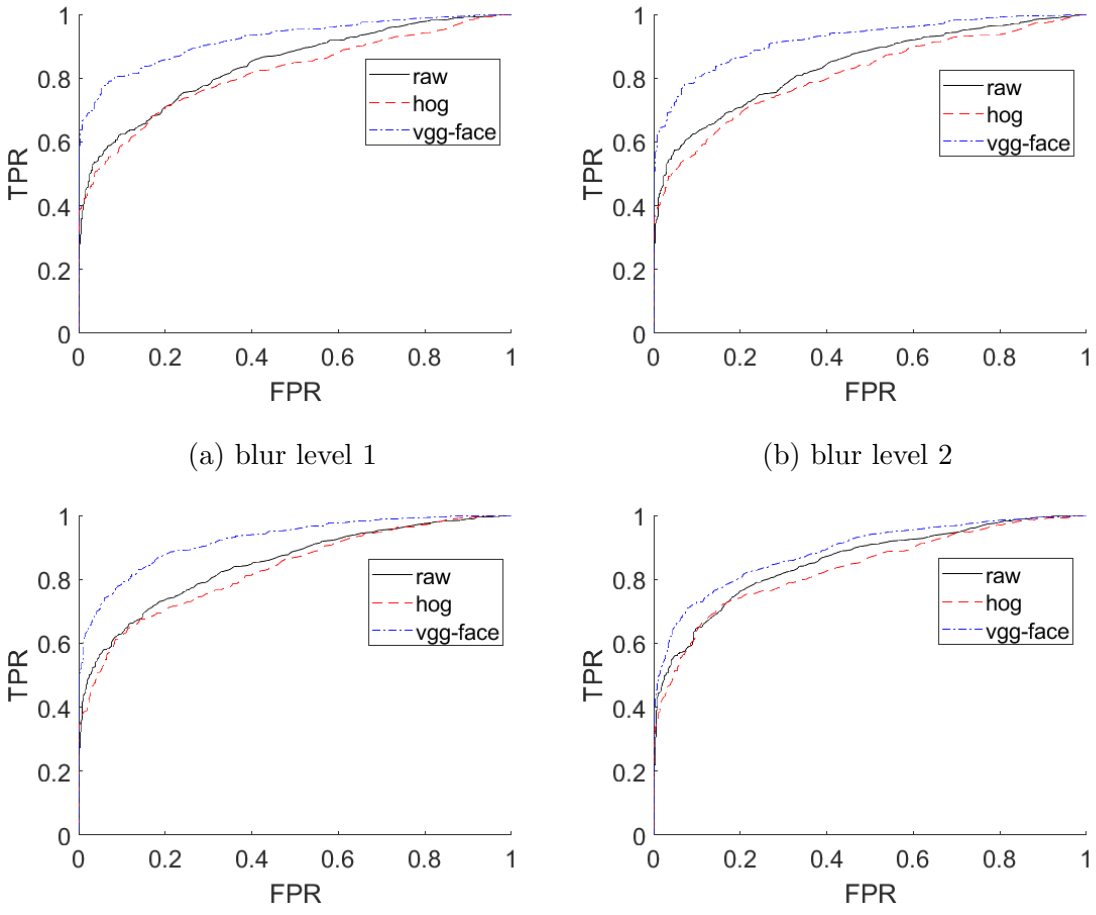
(c) blur level 3

(d) blur level 4

Figure 3.11: ROC curves of raw, HoG and VGG-Face for the Extended Yale B dataset. The test samples are blurred at (a) level 1 (b) level 2 (c) level 3 and (d) level 4. Different colors and symbols represent the different feature types. The ROC curves show that the HoG feature better separates the classes by providing the highest AUC.

are the largest for VGG-Face. However, while the FSCCI shows that the raw images feature performs better than HoG, the mean SCI does not follow the same trend. This proves one more time that the proposed FSCCI is more accurate for feature selection than the mean SCI. All FSCCI results are confirmed by the recognition accuracy values in Table 3.3 for the same considered feature types.

In the proposed framework, for a given application, the proposed FSCCI can be used first to determine the best performing feature based on the training set for that



(a) blur level 1

(b) blur level 2

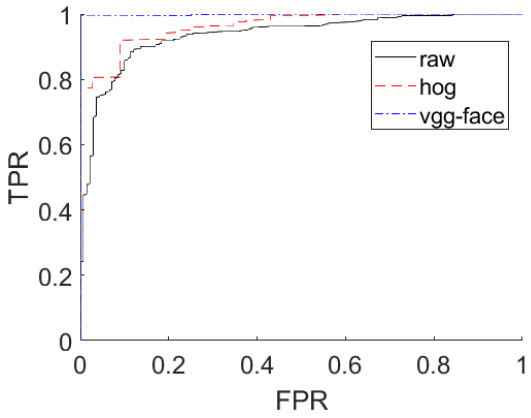
(c) blur level 3

(d) blur level 4

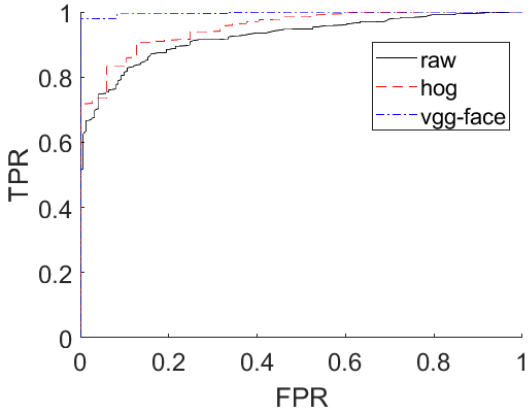
Figure 3.12: ROC curves of raw, HoG and VGG-Face for the LFW dataset. The test samples are blurred at (a) level 1 (b) level 2 (c) level 3 and (d) level 4. Different colors and symbols represent the different feature types. The ROC curves show that the VGG-Face feature better separates the classes by providing the highest AUC.

application.

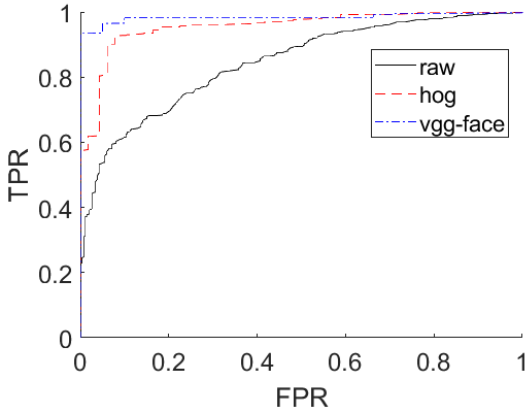
For a better visualization of the FSCCI score, the ROC curves are displayed for each feature type and blur level where the three different datasets are considered in turn. Fig. 3.10, Fig. 3.11 and Fig. 3.12 show the ROC curves for the ORL, Extended Yale B and LFW datasets, respectively. The ROC curves comply with the FSCCI results in Table 3.2 where the highest values correspond to VGG-Face (ORL), HoG (Extended Yale B) and VGG-Face (LFW). Similarly, the ROC curves are displayed



(a) occlusion size 10%

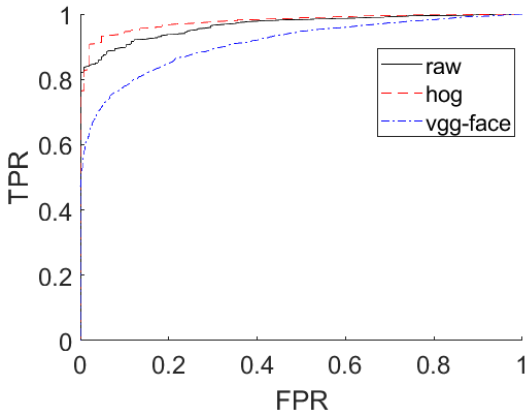


(b) occlusion size 15%

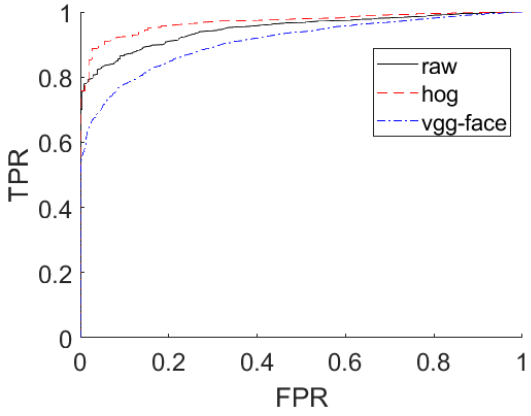


(c) occlusion size 25%

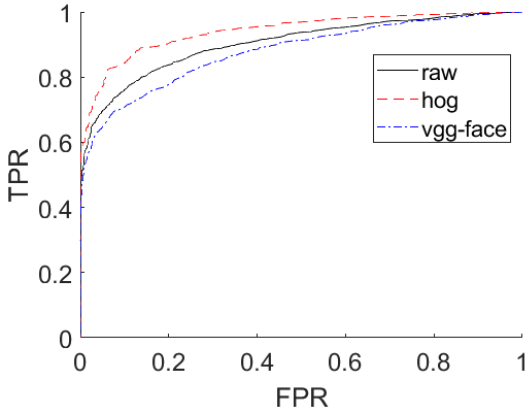
Figure 3.13: ROC curves of raw, HoG and VGG-Face for the ORL dataset. The test samples are occluded at occlusion (a) size 10% (b) size 15% and (c) size 25%. Different colors and symbols represent the different feature types. The ROC curves show that the VGG-Face feature better separates the classes by providing the highest AUC.



(a) occlusion size 10%

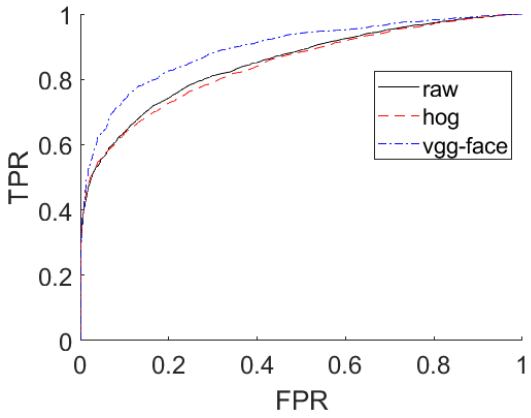


(b) occlusion size 15%

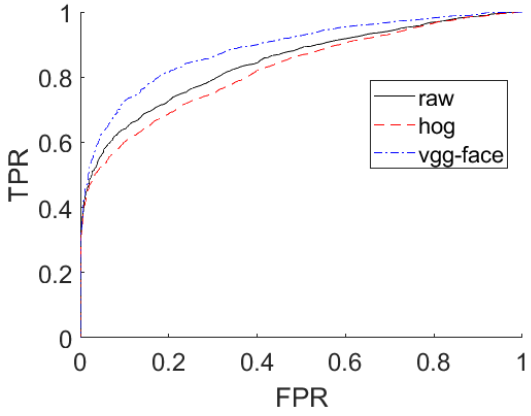


(c) occlusion size 25%

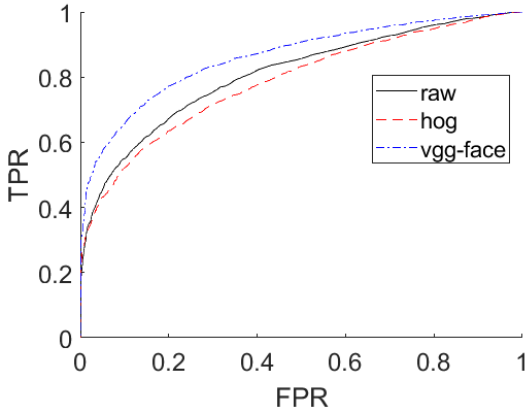
Figure 3.14: ROC curves of raw, HoG and VGG-Face for the Extended Yale B dataset. The test samples are occluded at occlusion (a) size 10% (b) size 15% and (c) size 25%. Different colors and symbols represent the different feature types. The ROC curves show that the HoG feature better separates the classes by providing the highest AUC.



(a) occlusion size 10%



(b) occlusion size 15%



(c) occlusion size 25%

Figure 3.15: ROC curves of raw, HoG and VGG-Face for the LFW dataset. The test samples are occluded at occlusion (a) size 10% (b) size 15% and (c) 25%. Different colors and symbols represent the different feature types. The ROC curves show that the VGG-Face feature better separates the classes by providing the highest AUC.

Table 3.4: Pearson Correlation Coefficient (PCC) of FSCCI and Mean SCI with Respect to Recognition Accuracy for the ASRC Framework using Raw, HoG and VGG-Face Features on the ORL, Extended Yale B and LFW Datasets. The Results are Displayed for the Blur and Occlusion Distortions. Bold Entries show the Highest Values for the PCC for Each Feature Type in a Dataset and for Each Distortion.

		PCC of FSCCI		**PCC of Mean SCI**	
Dataset	**Method**	Blur	Occlusion	Blur	Occlusion
ORL	ASRC+raw	**0.9300**	**0.9998**	0.7925	0.9802
	ASRC+HoG	**0.7998**	**0.9882**	0.5335	0.9880
	ASRC+VGG-Face	**0.9827**	**0.9996**	0.4495	0.9761
Extended Yale B	ASRC+raw	0.9896	**0.9930**	**0.9982**	0.9927
	ASRC+HoG	**0.8144**	**0.9921**	0.4454	0.9888
	ASRC+VGG-Face	0.9867	0.9716	**0.9985**	**0.9992**
LFW	ASRC+raw	0.9821	0.9757	**0.9955**	**0.9898**
	ASRC+HoG	**0.9967**	**0.9986**	0.5467	0.8201
	ASRC+VGG-Face	0.9918	0.9809	**0.9959**	**0.9979**

for each feature type and occlusion level for the three considered datasets. Again, Fig. 3.13, Fig. 3.14 and Fig. 3.15 show the same trend as with the Gaussian blur and comply with the FSCCI results in Table 3.3.

Finally, in order to evaluate the effectiveness of the proposed FSCCI in measuring the ability of a feature to preserve the sparsity and accuracy of the representation, the Pearson correlation coefficient (PCC) is calculated for the FSCCI with respect to the obtained recognition accuracy. Similarly, to compare the effectiveness of the FSCCI and mean SCI at relating to the feature performance, the PCC is calculated for the mean SCI with respect to the obtained recognition accuracy.

Table 3.4 displays the PCC of the FSCCI and mean SCI for the ASRC framework using raw, HoG and VGG-Face features on the ORL, Extended Yale B and LFW

datasets. The results are displayed for the blur and occlusion distortions. It is shown that the FSCCI is more correlated than the mean SCI to the recognition accuracy where higher values of the PCC are observed for FSCCI for almost all the considered cases. It is also worth noting that for the few cases where the PCC is lower for the FCSSI as compared to the mean SCI, the obtained PCC values are very close, which indicates similar correlation of both metrics with respect to the recognition accuracy.

3.4.4.2 ASRC Evaluation under Gaussian Blur

The results of the proposed ASRC framework are analyzed with respect to the conventional SRC framework, in addition to other state-of-the-art sparse-based methods and blur invariant methods, in the presence of different levels of Gaussian blur. The recognition accuracy rates are computed for the considered models when tested on the ORL, Extended Yale B and LFW datasets at four different Gaussian blur levels, varying from an imperceptible level (1) to a highly impaired level (4). The corresponding results are listed in Table 3.5

Comparison with Sparse-Based Methods: For the ORL dataset, when considering the raw, HoG and VGG-Face features, it is observed that the ASRC framework shows a better performance than the SRC and other sparse-based methods for all blur levels. This indicates that the proposed ASRC is capable of improving the SRC sparsity if the SRC representation is naturally sparse. In particular, the ASRC outperforms all other sparse-based methods at the highest blur level with all three considered features, as reported by the recognition accuracies in Table 3.5. For the Extended Yale B, the ASRC exhibits a consistent higher recognition accuracy over the SRC framework. When comparing the SRC and ASRC frameworks, the latter achieves higher recognition accuracies when used with all three features at all blur levels. It is worth noting that the performance of the ASRC is the highest when

used with the HoG feature. Similarly, it is observed that the ASRC outperforms other sparse-based methods at all blur levels, when it is particularly used with HoG features. For the unconstrained LFW dataset, it can be seen from Table 3.5 that the raw features are not discriminative enough to generate a sparse representation resulting in a poor recognition performance. This is clearly reported by the low recognition accuracy values for both SRC and ASRC. In Table 3.5, a similar trend is observed with the HoG features whose low mean SCI and FSCCI values (Table 3.2) reflect the inability to preserve the sparsity of the representation, which in turn leads to a poor recognition performance. From Table 3.2, it can be seen that the deep VGG-Face features are more robust and provide a sparser representation as reflected by the mean SCI and FSCCI values. Furthermore, Table 3.5 shows that while the conventional SRC (SRC+raw) results in a poor recognition performance at all blur levels, replacing the raw features with the feature corresponding to the highest FSCCI (VGG-Face feature from Table 3.2) leads to a significant improvement in recognition performance at lower blur levels (levels 1 and 2). At higher blur levels, the observed accuracy decrease with VGG-Face is alleviated by the ASRC, which shows a better performance as compared to the SRC. Furthermore, the ASRC exhibits a more consistent performance across all blur levels. From Table 3.5, it can be seen that, while the SRC recognition accuracy decreases by 54% between blur levels 1 and 4, the ASRC recognition accuracy decreases only by 26%. Similarly, the ASRC outperforms the other sparse-based methods at all blur levels when it is used with VGG-Face features.

Based on these results, the ASRC demonstrates an improvement over the SRC and the other sparse-based methods on the three different datasets when particularly used with the HoG and VGG-Face features in the presence of high levels of Gaussian blur. This proves that representing the images with features that are more resilient

to image variations, as compared to raw pixels, is harnessed by the ASRC framework structure that makes these features resilient to blur as well.

Comparison with Blur-Invariant Methods: In Table 3.5, it is observed that for the ORL dataset, the ASRC performs better than other blur-invariant methods when used with the HoG and deep features. For the Extended Yale B dataset, the ASRC outperforms all the other methods when used with raw and HoG features at all blur levels. Finally, for the LFW dataset, the ASRC proves to be better than the other blur invariant methods, when used with VGG-Face features.

It is worth noting that Table 3.5 shows that the ASRC performs the best with all features and across all blur levels, as compared to the existing ID [35] and Zhang's [34] distance methods. These two blur invariant methods are global methods that are originally designed to be invariant to blur but not to scene variations challenges. Their serious drawback is the fact that a local change of image affects the values of their invariant feature vectors. This is why global invariants cannot be used when the face is partially occluded or its appearance varies widely. However, the rDRBF performs better than ASRC with raw features when applied on the ORL and LFW datasets. This is mainly due to the fact that the rDRBF uses the local LBP features. Compared to raw pixels, LBP is a powerful texture operator and a robust approach to describing local structures and thus, is better suited to the face recognition uncontrolled challenges. Nevertheless, the ASRC with raw features performs better than rDRBF on the Extended Yale B, as the latter method is affected by the extreme illumination changes of the dataset.

3.4.4.3 ASRC Evaluation under Realistic Camera Shake Blur

In the next series of experiments, the proposed ASRC framework is evaluated on realistic blurred images resulting from camera shake, as described in Section 3.4.3.

Table 3.5: Recognition Accuracy (%) of the Proposed ASRC Method and Comparison with Sparse-Based Methods and Blur-Invariant Methods Under Different Levels of Gaussian Blur on ORL, Extended Yale B and LFW Datasets. Bold Entries Are the Best Performers for Each Blur Level in Each Dataset.

	ORL					**Extended Yale B**					**LFW**				
	Blur level					Blur level					Blur level				
	level 1	level 2	level 3	level 4	Average	level 1	level 2	level 3	level 4	Average	level 1	level 2	level 3	level 4	Average
Sparse-Based Method															
RLSI [20]	90.50	90.50	91.50	91.50	91.00	97.26	97.10	96.78	95.66	96.70	63.47	63.14	58.77	52.75	59.53
ESRC [22]	91.94	91.94	91.35	91.40	91.65	98.31	97.99	97.43	96.24	97.49	60.95	60.75	54.53	44.28	55.12
RSC [18]	90.50	89.50	89.00	90.50	89.87	96.78	96.94	96.78	96.30	96.70	64.06	64.13	59.83	49.24	59.31
SRC [23]	92.50	93.00	92.00	92.00	92.37	94.93	95.17	94.85	94.05	94.75	50.69	49.77	45.53	42.62	47.15
SRC+HoG	98.00	98.00	97.00	94.50	96.87	**98.95**	**98.71**	97.99	96.86	98.12	37.46	36.60	35.94	30.05	35.01
SRC+VGG-Face	**100.0**	**100.0**	**100.0**	96.50	99.12	82.46	80.29	76.59	69.99	77.33	**82.53**	**85.44**	69.23	37.79	68.74
ASRC+raw	92.50	93.00	93.50	93.50	93.12	98.71	98.23	97.35	95.82	97.52	45.40	44.41	41.76	38.19	42.44
ASRC+HoG	97.00	96.50	96.00	97.50	96.75	98.79	**98.71**	**98.23**	**97.59**	**98.33**	38.12	38.39	36.14	35.21	36.96
ASRC+VGG-Face	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	81.01	80.93	78.60	74.09	78.65	81.01	80.93	**78.60**	**74.09**	**73.87**
Blur-Invariant Method															
ID [35]	79.00	77.50	74.50	76.00	76.75	49.24	49.32	48.51	48.19	48.81	34.75	35.14	34.15	32.53	34.14
Zhang's distance [49]	85.75	86.50	84.25	80.75	84.31	55.99	55.99	54.71	53.66	55.08	31.51	21.31	21.31	20.35	23.62
rDRBF [36]	95.50	96.50	96.00	97.00	96.25	86.56	86.00	85.52	84.55	85.65	60.62	61.02	58.84	55.86	59.08

Table 3.6: Recognition Accuracy (%) of ASRCra raand SRC under 8 Different Levels of Realistic Blur [1] on the ORL Dataset. GB-ASRC Corresponds to the ASRC Dictionary Augmented with Gaussian Blurred Images and RB-ASRC Corresponds to the ASRC Dictionary Augmented with Realistic Blurred Images. Bold Entries Are the Best Performers.

	Blur level								
Method	level 1	level 2	level 3	level 4	level 5	level 6	level 7	level 8	Average
SRC [23]+raw	90.50	91.50	90.50	91.50	91.50	88.00	87.00	80.00	88.81
SRC+HoG	97.00	97.50	94.50	96.00	96.00	93.00	89.50	84.00	93.43
SRC+VGG-Face	**100.0**	**100.0**	48.50	86.00	86.00	77.50	42.00	11.00	68.87
GB-ASRC+raw	91.50	91.50	92.50	92.00	92.00	90.00	92.00	83.50	90.62
GB-ASRC+HoG	96.00	96.00	96.50	96.00	96.00	96.00	94.00	88.00	94.81
GB-ASRC+VGG-Face	**100.0**	**100.0**	98.50	**100.0**	**100.0**	96.50	86.00	68.00	93.62
RB-ASRC+raw	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	94.50	95.00	90.50	97.50
RB-ASRC+HoG	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	98.50	**96.00**	**99.31**
RB-ASRC+VGG-Face	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	99.50	**100.0**	92.50	99.00

Table 3.7: Recognition Accuracy (%) of ASRC and SRC under 8 Different Levels of Realistic Blur [1] on the Extended Yale B dataset. GB-ASRC Corresponds to the ASRC Dictionary Augmented with Gaussian Blurred Images and RB-ASRC Corresponds to the ASRC Dictionary Augmented with Realistic Blurred Images. Bold Entries Are the Best Performers.

	Blur level								
Method	level 1	level 2	level 3	level 4	level 5	level 6	level 7	level 8	Average
SRC [23]+raw	94.61	94.85	93.24	93.89	74.74	84.55	80.29	61.14	84.66
SRC+HoG	98.23	98.39	95.74	96.70	94.29	93.40	91.07	84.15	93.99
SRC+VGG-Face	78.28	77.96	59.69	65.25	68.82	54.63	42.24	39.18	60.75
GB-ASRC+raw	97.43	97.91	94.53	95.49	81.09	86.00	82.94	69.27	88.08
GB-ASRC+HoG	**98.31**	**98.31**	97.02	97.18	94.21	93.89	92.36	86.08	94.67
GB-ASRC+VGG-Face	78.76	79.49	66.45	71.20	71.60	59.53	49.88	48.67	65.69
RB-ASRC+raw	97.59	97.75	94.21	94.93	96.22	**94.05**	**92.92**	92.76	95.05
RB-ASRC+HoG	98.23	98.23	**97.10**	**97.51**	**96.46**	**94.05**	**92.92**	**92.84**	**95.91**
RB-ASRC+VGG-Face	79.49	79.32	69.59	71.52	71.92	61.38	55.43	58.97	68.45

The performance of the proposed ASRC framework is demonstrated with respect to the conventional SRC in the presence of eight different levels of blur distortion [1]. Two different cases are explored for the ASRC dictionary structure. In the first case, the GB-ASRC augments the ASRC dictionary with Gaussian blurred images at the same four distortion levels as described in Section 3.4.4.2. In the second case, the RB-ASRC augments, this time, the ASRC dictionary with realistic blurred images resulting from four of the 8 blur kernels of Levin *et al.* [1]. The sizes of the four selected blur kernels are 13×13, 17×17, 21×21 and 27×27. The recognition accuracy results are displayed in Table 3.6, Table 3.7 and Table 3.8 for the ORL, Extended Yale B and LFW datasets, respectively.

For the ORL dataset, when used with the SRC, the deep features show a sharp decrease in performance at high blur levels where they are outperformed by the HoG

Table 3.8: Recognition Accuracy (%) of ASRC and SRC under 8 Different Levels of Realistic Blur [1] on the LFW Dataset. GB-ASRC Corresponds to the ASRC Dictionary Augmented with Gaussian Blurred Images and RB-ASRC Corresponds to the ASRC Dictionary Augmented with Realistic Blurred Images. Bold Entries Are the Best Performers.

	Blur level								
Method	level 1	level 2	level 3	level 4	level 5	level 6	level 7	level 8	Average
SRC [23]+raw	60.69	58.31	51.29	54.93	21.64	31.04	26.41	13.63	40.84
SRC+HoG	58.23	54.78	47.65	49.82	18.51	27.38	23.67	11.21	36.40
SRC+VGG-Face	83.92	77.04	68.36	41.96	60.62	40.83	25.94	10.06	51.84
GB-ASRC+raw	57.25	54.60	47.58	49.11	27.86	34.08	32.30	20.32	40.38
GB-ASRC+HoG	56.77	52.34	45.18	46.71	25.14	31.42	30.06	18.16	38.22
GB-ASRC+VGG-Face	**83.98**	**79.48**	**72.41**	**59.96**	63.07	45.93	31.11	22.90	57.32
RB-ASRC+raw	52.42	51.22	47.78	49.83	45.47	44.41	45.20	41.89	47.24
RB-ASRC+HoG	50.18	49.83	46.58	47.78	44.12	41.07	40.16	38.19	44.73
RB-ASRC+VGG-Face	82.06	77.96	70.59	59.10	**68.30**	**50.89**	**41.10**	**40.44**	**61.30**

and raw images features. However, when used with the ASRC, the deep features demonstrate a noticeable amelioration in terms of recognition accuracy at the same high blur levels, especially in the case of the RB-ASRC, where the dictionary is augmented with the same type of blur as in the test samples. In this case, VGG-Face and HoG features have a similar high performance at all blur levels (Table 3.6).

For the Extended Yale B, the HoG features prove to have a consistent performance at all blur levels compared with raw images and VGG-Face features. However, the best performance is achieved with the ASRC framework, especially, the RB-ASRC, as shown in Table 3.7.

Finally, being an unconstrained dataset, the LFW proves once again that the choice of features is major in providing a good recognition accuracy when used with SRC and ASRC. Therefore, the VGG-Face features show that they have the best

Table 3.9: Recognition Accuracy (%) of SRC and ASRC under Different Sizes of Single Block Occlusion on the ORL, Extended Yale B and LFW Datasets. Bold Entries Are the Best Performers for each Occlusion Size in Each Dataset.

	ORL				**Extended Yale B**				**LFW**			
	Occlusion size %				Occlusion size %				Occlusion size %			
Method	10	15	25	Average	10	15	25	Average	10	15	25	Average
SRC [23]+raw	83.75	78.00	62.25	74.66	95.07	92.27	78.04	88.46	44.32	34.29	18.33	32.31
SRC+HoG	96.75	95.62	92.25	94.87	96.68	95.81	93.30	95.26	39.03	33.22	25.48	32.57
SRC+VGG-Face	99.62	97.62	89.25	95.50	66.13	62.47	55.49	61.36	66.77	58.27	42.26	55.77
ASRC+raw	85.50	83.37	71.00	79.95	96.68	94.97	86.49	92.71	42.53	39.21	32.94	38.23
ASRC+HoG	96.37	96.00	94.37	95.58	**98.27**	**97.91**	**94.69**	**96.95**	36.68	32.54	26.36	31.86
ASRC+VGG-Face	**99.50**	**99.37**	**98.50**	**99.12**	77.03	73.39	65.06	71.82	**71.04**	**67.09**	**58.23**	**65.45**

performance in this case. However, this type of features is easily affected by the presence of high blur levels, especially when used with the SRC. Their performance is, nonetheless, largely improved when used with the ASRC as shown in Table 3.8.

It is worth noting that both ASRC structures (GB-ASRC and RB-ASRC) outperform the SRC framework for all three feature types, especially at high blur levels. Thus, although the ASRC performs the best at high blur levels when the same type of blur is used in the dictionary as in the test samples, the ASRC maintains a sparser representation than the SRC regardless of the blur type.

3.4.4.4 ASRC Evaluation under Block Occlusion

In this series of experiments, the proposed ASRC framework is evaluated in the presence of different sizes and locations of block occlusion. The performance of the proposed ASRC framework is demonstrated with respect to the conventional SRC in the presence of single and double occlusions.

Fig. 3.16 to Fig. 3.18 show the recognition rate across the entire range of single occlusion positions for various feature types for the ORL, Extended Yale B and LFW datasets, respectively. For this purpose, the block occlusion size in the test images is

Table 3.10: Average Recognition Accuracy (%) of SRC and ASRC under Different Sizes of Double Block Occlusions on the ORL, Extended Yale B and LFW Datasets. Bold Entries Are the Best Performers for Each Occlusion Size in Each Dataset.

	ORL				**Extended Yale B**				**LFW**			
	Occlusion size %				Occlusion size %				Occlusion size %			
Method	30	40	50	Average	30	40	50	Average	30	40	50	Average
SRC [23]+raw	69.50	62.75	45.50	59.25	89.01	75.74	42.11	68.95	29.05	13.63	12.21	18.29
SRC+HoG	94.50	92.00	87.50	91.30	92.39	85.80	73.65	83.94	27.23	18.89	17.93	21.35
SRC+VGG-Face	86.25	75.50	63.00	74.91	44.41	38.66	30.61	37.89	34.68	28.06	19.55	28.03
ASRC+raw	80.75	74.25	53.75	69.58	95.33	84.71	45.05	75.03	37.26	21.97	18.96	26.06
ASRC+HoG	96.00	**95.50**	**91.00**	**94.16**	**97.22**	**94.81**	**84.19**	**92.07**	32.40	20.38	17.57	23.45
ASRC+VGG-Face	**99.25**	91.00	73.25	87.83	70.15	64.24	48.87	61.09	**56.58**	**49.33**	**40.07**	**48.66**

varied from 10 percent to 25 percent of the image size at four different positions, as illustrated in Fig. 3.8(a). For the double occlusion, the total block occlusion is varied from 30 percent to 50 percent of the image size at the same four different positions as illustrated in Fig. 3.8(b). Tables 3.9 and 3.10 show the average recognition rate for single occlusion and for double occlusion.

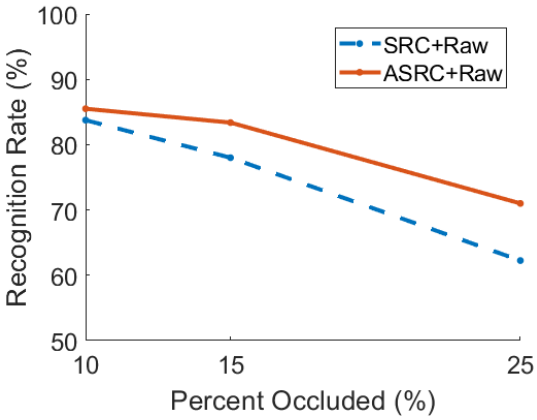
For the ORL dataset, the single occlusion recognition rate is the highest for ASRC, as compared to SRC, when used with raw pixels and VGG-Face features at all sizes of occlusion (Table 3.9 and Fig. 3.16). When used with HoG, the ASRC and SRC accuracies are close in values for less than the 25 percent occlusion size. A sharp decrease in accuracy of 10.75% is specifically observed between the 10 percent occlusion size and the 25 percent occlusion size when using SRC with VGG-Face. In contrast, the accuracy remains steady with a variation of only 1.5% between the 10 percent occlusion size and the 25 percent occlusion size when using ASRC with VGG-Face. For the double occlusion average recognition rate (Table 3.10), it is observed again that ASRC outperforms SRC for all feature types and occlusion sizes. For the Extended Yale B dataset, the ASRC demonstrates a consistent superiority in its performance with respect to the SRC when considering the three different features at all occlusion

sizes. The ASRC with the HoG feature achieves the best recognition accuracies for all occlusion sizes (Table 3.9 and Fig. 3.17). For the double occlusion, the ASRC follows again the same trend as the single occlusion case, where it performs the best with the HoG features (Table 3.10). For the LFW dataset, for the single occlusion case, it can be seen from Table 3.9 and Fig. 3.18, that the proposed ASRC+VGG-Face consistently outperforms SRC for all occlusion sizes. In addition, the proposed ASRC exhibits a smoother degradation in performance as compared to SRC. From Table 3.9, it is observed a decrease in accuracy of 77.39% for SRC as compared to only 41.2% for ASRC between the 10 percent occlusion size and the 25 percent occlusion size. Table 3.10 shows the results for the double occlusion case, where the ASRC proves to be again better than SRC at all occlusion levels. The VGG-Face features perform the best due to the unconstrained nature of the LFW images, where the raw and HoG features are not discriminative enough.

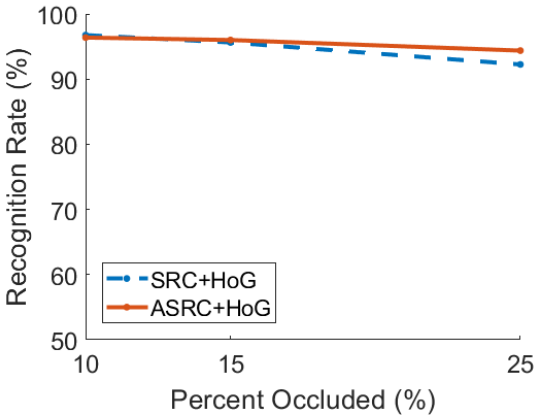
The obtained results show that the ASRC model shows a significant improvement over the SRC framework when used with features, such as HoG and VGG-Face, in the presence of large sizes of occlusion exceeding 30 percent. This proves that the proposed model, which represents the occlusion distortion with a linear pointwise multiplication operation instead of an additive operation, can handle better block occlusions covering more than 33 percent of the image size. Moreover, while the VGG-Face features perform poorly with the SRC in the presence of large occlusion sizes, they exhibit a higher robustness to occlusion when used with the ASRC framework.

Table 3.11 shows the recognition rate for single occlusion at random positions for various feature types on the ORL, Extended Yale B and LFW datasets, respectively. For this purpose, the block occlusion size in the test images is varied from 15 percent to 35 percent of the image size at random different positions.

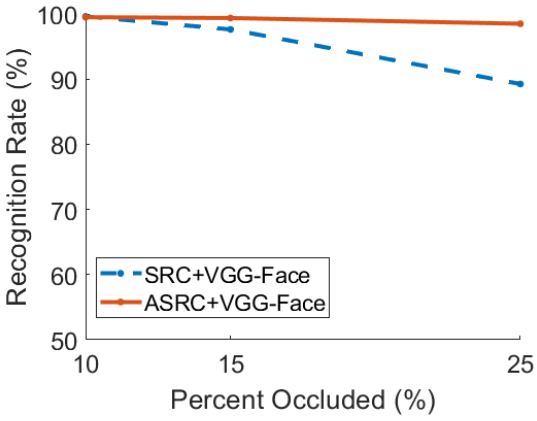
For the ORL dataset, the occlusion recognition rate is the highest for ASRC, as



(a) Raw

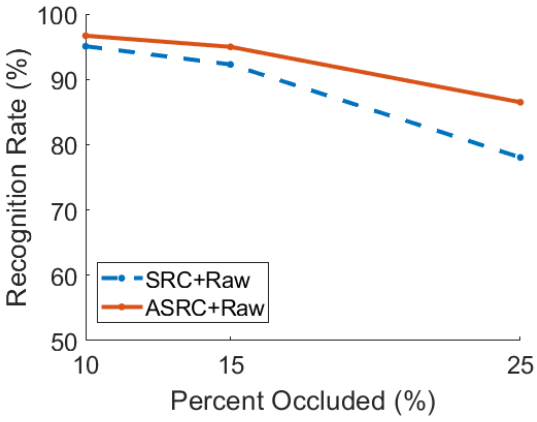


(b) HoG

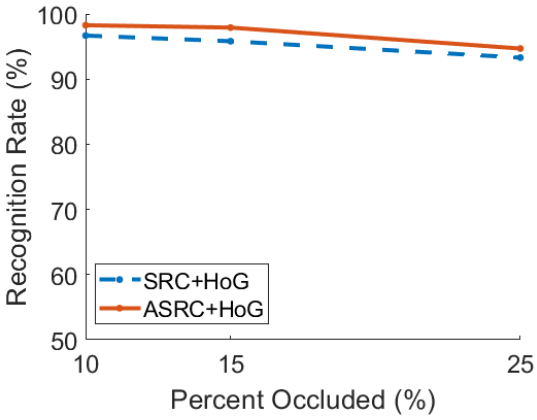


(c) VGG-Face

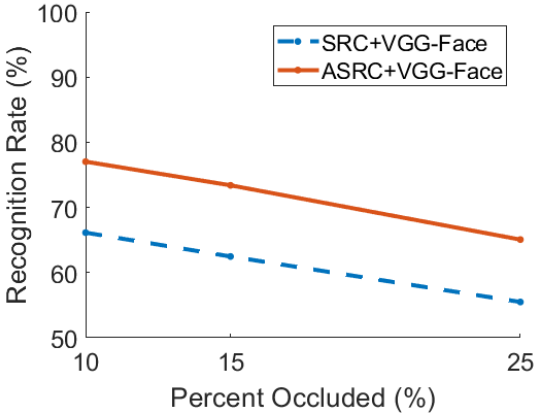
Figure 3.16: Recognition rate of SRC and ASRC for the ORL dataset where the occlusion block size varies from 10% to 25% of the image size. Performance is evaluated for different feature types (a) Raw, (b) HoG and (c) VGG-Face. The performance curves show that the ASRC better recognizes the occluded faces.



(a) Raw

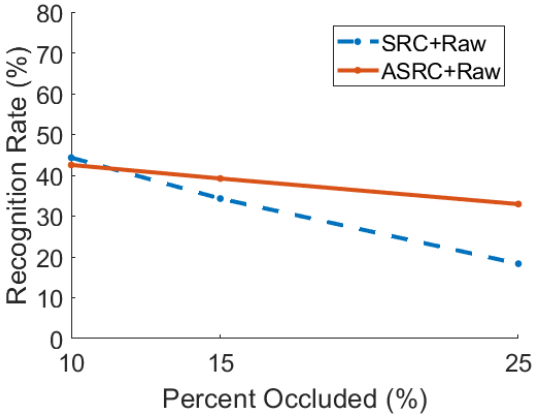


(b) HoG

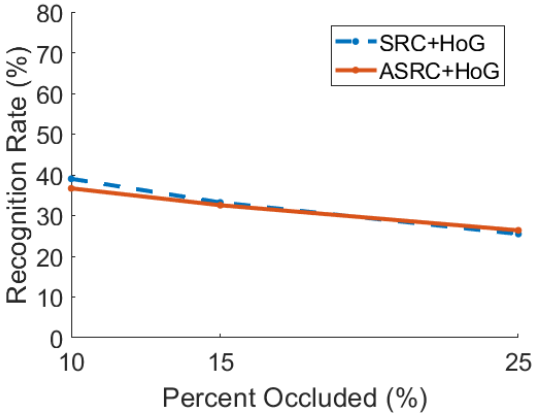


(c) VGG-Face

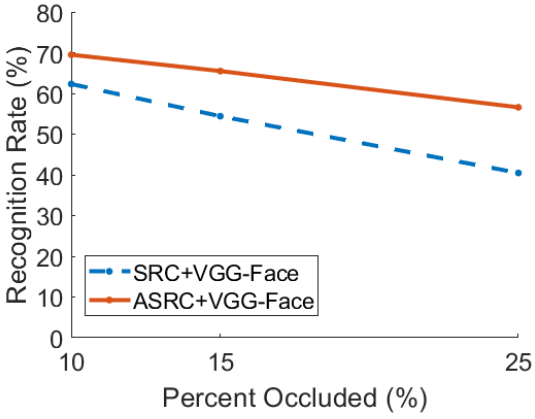
Figure 3.17: Recognition rate of SRC and ASRC for the Extended Yale B dataset where the occlusion block size varies from 10% to 25% of the image size. Performance is evaluated for different feature types (a) Raw, (b) HoG and (c) VGG-Face. The performance curves show that the ASRC better recognizes the occluded faces.



(a) Raw



(b) HoG



(c) VGG-Face

Figure 3.18: Recognition rate of SRC and ASRC for the LFW dataset where the occlusion block size varies from 10% to 25% of the image size. Performance is evaluated for different feature types (a) Raw, (b) HoG and (c) VGG-Face. The performance curves show that the ASRC better recognizes the occluded faces.

Table 3.11: Recognition Accuracy (%) of SRC and ASRC under Different Sizes of Single Block Occlusion at Random Positions on the ORL, Extended Yale B and LFW Datasets. Bold Entries Are the Best Performers for Each Occlusion Size in Each Dataset.

	ORL				**Extended Yale B**				**LFW**			
	Occlusion size %				Occlusion size %				Occlusion size %			
Method	15	25	35	Average	15	25	35	Average	15	25	35	Average
SRC [23]	70.50	57.00	20.00	49.16	96.00	79.57	65.73	80.40	30.44	16.48	07.08	18.00
SRC+HoG	96.50	94.50	70.50	**87.16**	95.22	93.89	87.45	92.18	33.82	25.22	16.41	25.15
SRC+VGG-Face	98.50	93.00	28.50	73.33	63.07	48.91	34.43	48.80	59.89	40.04	16.08	38.67
ASRC+raw	69.50	61.00	23.00	51.16	91.00	83.99	73.93	82.97	32.96	26.80	15.49	25.08
ASRC+HoG	93.00	92.50	**73.00**	86.16	**96.22**	**94.37**	**89.78**	**93.45**	29.19	23.10	16.81	23.03
ASRC+VGG-Face	**100.0**	**99.00**	56.50	85.16	71.52	63.72	54.95	63.39	**71.04**	**67.09**	**58.23**	**51.51**

compared to SRC, when used with raw pixels and VGG-Face features at all sizes of occlusion (Table 3.11). When used with HoG, the ASRC and SRC accuracies are close in values for less than the 25 percent occlusion size. A sharp decrease in accuracy of 64.91% is specifically observed between the 15 percent occlusion size and the 25 percent occlusion size when using SRC with VGG-Face. In contrast, the accuracy decreases only by 1% between the 15 percent occlusion size and the 25 percent occlusion size when using ASRC with VGG-Face.

For the Extended Yale B dataset, the ASRC demonstrates a consistent superiority in its performance with respect to the SRC when considering the three different features at all occlusion sizes. The ASRC with the HoG feature achieves the best recognition accuracies for all occlusion sizes (Table 3.11).

For the LFW dataset, it can be seen from Table 3.11, that the proposed ASRC+VGG-Face consistently outperforms SRC for all occlusion sizes. In addition, the proposed ASRC exhibits a smoother degradation in performance as compared to SRC. From Table 3.11, it is observed that a decrease in accuracy of 76.74% occurs for SRC as compared to only 11.37% for ASRC between the 10 percent occlusion size and the

Table 3.12: Average Recognition Accuracy (%) of SRC and ASRC under Different Sizes of Single Block Occlusion at Random Positions on the ORL Dataset for 25 Iterations. Bold Entries Are the Best Performers for Each Occlusion Size.

	Occlusion size %			
Method	15	25	35	Average
SRC [23]	73.13	53.58	32.95	53.22
SRC+HoG	96.37	93.55	88.62	92.84
SRC+VGG-Face	99.23	92.67	72.42	88.10
ASRC+raw	69.03	63.80	49.50	60.77
ASRC+HoG	93.65	91.95	90.73	92.11
ASRC+VGG-Face	**99.85**	**97.97**	**87.17**	**94.99**

35 percent occlusion size. The VGG-Face features perform the best due to the unconstrained nature of the LFW images, where the raw and HoG features are not discriminative enough.

To improve the accuracy of the previous results, the same series of experiments is repeated for the ORL dataset, where the proposed ASRC framework is evaluated in the presence of the same sizes of random block occlusion for 25 iterations instead of one. Table 3.12 shows the average recognition rate for single occlusions at random positions for the ORL dataset. Therefore, the block occlusion size in the test images is varied from 15 percent to 35 percent of the image size at random different positions and the average recognition rate is considered over 25 different experimental test sets.

It can be noted that the occlusion recognition rate is the highest for ASRC, as compared to SRC, when used with raw pixels at 25% and 35% occlusion sizes (Table 3.12). When used with HoG, the ASRC and SRC accuracies are close in values for less than the 35 percent occlusion size. A large improvement in accuracy is specifically observed for the VGG-Face feature for both SRC and ASRC compared with the previous results in Table 3.11. Moreover, the ASRC outperforms the SRC for all occlusions sizes by up to 20% for the 35% occlusion size.

3.4.4.5 ASRC Evaluation under Additive White Noise

In the last set of experiments, the results of the proposed ASRC framework are analyzed in the presence of different levels of white noise. As described in Section 3.3.2.2 and Algorithm 2, the non-sparse white noise case is handled by first applying a lowpass filter to the noisy images. The lowpass filtered images are then input to the proposed ASRC. In the implementation, a 5×5 Gaussian lowpass filter with a variance of 4 is applied to the noisy test images.

The performance of the proposed ASRC framework is first demonstrated with respect to the conventional SRC in the presence of white noise. Table 3.13 shows the recognition accuracy rate for the SRC and ASRC frameworks for different features, when the white noise level varies from an imperceptible level (1) to a highly impaired level (4). For comparison, Table 3.13 also shows the recognition accuracy rate for existing sparse-based methods.

For the ORL dataset, the accuracy is the highest for ASRC as compared to SRC for all the considered features (raw pixels, HoG and VGG-Face) at high levels of white noise. At lower noise levels, the ASRC and SRC achieve both a high performance when the VGG-Face features are used. A sharp decrease in accuracy of 88.5% is specifically observed between noise level 1 and level 4 when using SRC with VGG-Face. In contrast, the accuracy remains steady with a variation of only 1% between level 1 and level 4 when using ASRC with VGG-Face. It is interesting to note that the proposed ASRC achieves a higher performance than SRC for all noise levels when raw pixels and HoG are used as features. Compared with other sparse-based methods, the ASRC performs the best with all features at all noise levels. For the Extended Yale B dataset, the ASRC demonstrates a consistent superiority in performance with respect to the SRC when considering the three different features. The ASRC with raw pixels

Table 3.13: Recognition Accuracy (%) of the Proposed ASRC Method and Comparison with Sparse-Based Methods under Different Levels of White Noise on ORL, Extended Yale B and LFW Datasets. Bold Entries Are the Best Performers for Each White Noise Level in Each Dataset.

	ORL					**Extended Yale B**					**LFW**				
	Noise level					Noise level					Noise level				
Method	level 1	level 2	level 3	level 4	Average	level 1	level 2	level 3	level 4	Average	level 1	level 2	level 3	level 4	Average
RLSI [20]	89.50	88.50	90.00	85.00	88.25	95.17	92.20	88.74	82.94	89.76	65.85	65.52	64.26	55.79	62.85
ESRC [22]	92.47	92.47	92.51	91.44	92.22	**97.51**	95.66	91.79	86.36	92.83	63.60	62.34	57.31	40.17	55.85
RSC [18]	91.00	91.50	89.00	90.00	90.37	95.74	93.31	92.04	90.27	92.84	65.45	66.31	64.66	55.92	63.08
SRC [23]	91.50	92.00	92.50	92.50	92.12	95.01	93.32	90.27	84.79	90.84	49.83	49.77	49.77	48.58	49.48
SRC+HoG	98.00	98.50	97.00	82.00	93.87	85.26	78.76	70.39	55.83	72.56	37.46	36.60	35.94	30.05	35.01
SRC+VGG-Face	**100.00**	**100.00**	99.50	11.50	77.75	72.65	67.58	55.75	21.16	54.28	**87.29**	**85.11**	66.98	10.72	62.52
ASRC+raw	93.50	93.50	93.00	92.50	93.12	97.43	**96.38**	**95.09**	**90.90**	**94.95**	45.53	45.47	44.34	44.41	44.93
ASRC+HoG	96.00	95.50	96.00	96.50	96.00	90.19	81.90	74.09	65.00	77.79	38.98	38.19	37.00	31.44	36.40
ASRC+VGG-Face	**100.00**	**100.00**	**100.00**	**99.00**	**99.75**	73.85	70.47	64.84	45.45	63.65	84.58	82.99	**80.68**	**66.51**	**78.69**

achieves the best recognition accuracies at all noise levels compared with SRC and the other sparse-based methods. It is worth noting that when used with VGG-Face, the SRC accuracy decreases by 70.47% between noise level 1 and level 4, while the ASRC accuracy decreases by only 38.45% between the same two levels. For the LFW dataset, as discussed in Section 3.4.4.2 and Section 3.4.4.4, the raw and HoG features cannot adequately sparsely represent the LFW dataset test images. Thus, using raw and HoG features with SRC or ASRC results in a poor recognition performance at all noise levels. When both frameworks are used with VGG-Face features, ASRC outperforms SRC, particularly at medium and high white noise levels. The ASRC also achieves a more consistent performance across all levels where a decrease in accuracy of 87.72% is observed for SRC as compared to only 21.36% for ASRC between noise level 1 and level 4. From Table 3.13, it can be seen that the ASRC outperforms other sparse-based methods when used with the VGG-Face features at all noise levels.

From the obtained results, one can conclude that the ASRC performance shows a significant improvement over the SRC framework when used with features that

preserve the sparsity of the representation in the presence of additive white noise.

3.5 Conclusions and Discussions

In this chapter, an Augmented SRC (ASRC) is proposed for face recognition under blur, non-sparse additive noise and block occlusion distortions. Since the ASRC framework represents the visual distortions using linear operations, augmenting its dictionary with these distortions makes the ASRC more robust to quality distortions than the conventional SRC by preserving the representation sparsity, when specifically used with discriminant features. While the blur distortion (Gaussian and camera shake) is represented by the proposed method using a linear convolution operation, the additive noise is represented using the same linear operation after converting the noise problem into a blur problem, and the block occlusion is represented using a linear pointwise vector multiplication. It is also showed that the feature space choice is important to enhance the performance of sparse representation-based classifiers. To aid in feature selection, a novel feature quality assessment index is presented, called Feature Sparsity Concentration and Classification Index (FSCCI) that is capable of assessing the feature quality in terms of both sparsity concentration and recognition accuracy. The ASRC sparse-based framework is evaluated on three constrained and unconstrained benchmark face datasets. In the evaluations, the raw images are used as features, in addition to the hand-crafted HoG and the deep VGG-Face. However, the ASRC framework can be also used with other features as well. The ASRC framework was shown to outperform popular sparse-based methods, including the conventional SRC,and blur-invariant methods, especially at medium to high distortion levels, and when particularly used with discriminative features, such as HoG and VGG-Face, as also validated by the proposed FSCCI.

Chapter 4

UNCONSTRAINED EAR RECOGNITION USING DEEP NEURAL NETWORKS FEATURES

The material covered in this chapter has been published in [11]. In this work, unconstrained ear recognition is performed using transfer learning with deep neural networks (DNNs). First, it is shown how existing DNNs can be used as a feature extractor. The extracted features are used by a shallow classifier to perform ear recognition. Performance can be improved by augmenting the training dataset with small image transformations. Next, the performance of the feature-extraction models is compared with fine-tuned networks. However, because the datasets are limited in size, a fine-tuned network tends to over-fit. Comparing with a deep learning based averaging ensemble that reduces the effect of over-fitting is proposed. Performance results are provided on unconstrained ear recognition datasets, the AWE and CVLE datasets as well as a combined AWE+CVLE dataset. It is shown that, in the case where long training time is not desirable or a large amount of data is not available, the features from pre-trained DNNs can be used with a shallow classifier to give a similar performance as fine-tuned networks.

4.1 Introduction

Accurate biometrics play a critical role in personal authentication and in forensic and security applications. A useful biometric modality has several desirable characteristics: uniqueness, ease of data collection, and preservation of privacy, among others. Uniqueness ensures that the biometric can be used to uniquely identify a person. Ease of data collection enables the biometric to be used in large scale surveil-

lance applications. Privacy preservation is increasingly important as many subjects may not want their personal identity easily accessible. Several biometrics meet these requirements to various degrees: face, iris, fingerprint, and ear. Face as a biometric meets the uniqueness and ease of collection criteria, but does not protect privacy. Iris as a biometric is unique and protects privacy, however may be difficult to collect. Fingerprints are unique and protect privacy, but also may be difficult to collect. This leaves us with ear, which is perhaps less often used than faces, but offers several unique advantages.

Just like a face or a fingerprint, the ear has a unique structure that can be used to identify the subject. However, compared with faces, the ear features are stable and are not affected by external factors, such as aging and expression. This is because the ear shape matures early in life and later changes occur gradually [113]. Compared with fingerprint recognition, ear recognition does not require the expensive capture of prints, and can be utilized in a visual surveillance application. Compared with iris recognition, ear recognition does not require subject cooperation. The main drawback of ear recognition is that the ear may be partially or fully occluded by hair, earrings, or other head-ware. However, it should be noted that face recognition has similar problems with occlusions due to glasses or head-ware. An additional benefit of ear recognition, instead of face recognition, is that there may be less privacy concerns when an image of an ear is captured and stored instead of an image of a face. Ears share more in common with fingerprints in that, although they have unique statistics that can be used to identify an individual, at a glance it is difficult for a human subject to recognize the identity using only the ear image.

Many approaches have been developed with the aim to improve ear detection and recognition capabilities for reliable deployment in surveillance and commercial applications [24, 25, 26, 27, 28, 29]. These approaches follow a traditional pipeline of

normalization, feature extraction and classification. In these works, the main challenge remains a proper selection of feature descriptors that can be resilient to unconstrained conditions, such as illumination changes, occlusion and quality distortions. More recent works (e.g. [30, 31]) use deep neural networks (DNNs) to end-to-end learn a classifier instead of designing a feature-classifier pipeline.

The use of DNNs as a feature extractor is explored in the more traditional feature-classifier pipeline approach. It is worth noting that features from pre-trained DNNs have been used in combination with shallow classifiers for a variety of computer vision tasks [32]. In this work, it is shown that features from pre-trained networks achieve a strong baseline for unconstrained ear recognition. Finally, they are compared with the performance of fine tuned deep networks that are expected to achieve greater performance.

This chapter is organized as follows. Section 4.2 discusses the related work on ear biometric recognition. Section 4.3 presents the proposed feature-based SVM model. Section 4.4 describes the experimental setup and results. Finally, Section 4.5 concludes the work.

4.2 Previous Work

Early ear recognition methods were structural methods based on physiological features such as shape, wrinkles, and ear points. The Iannarelli System of Ear Identification [7] was introduced in 1949 as one of the first systems to use the ear as a biometric modality for forensic science. The system consists of taking a certain number of measurements around the ear for a unique ear characterization. Much later, Moreno *et al.* [114] combined the results of several neural classifiers, which were trained on various ear geometrical features. Mu *et al.* [115] proposed an edge-based feature vector consisting of the ear's inner and outer structure and shape. Choras [58] computed

the centroid of ear curves to form concentric circles. Using the points between concentric circles and ear contours, two feature vectors were proposed. Later, Choras and Choras [59] added two more geometric feature vectors using representation of ear contours and a geometrical parametric method. Anwar *et al.* [28] proposed a method for ear recognition based on geometrical features extraction including shape, mean, centroid and Euclidean distance between pixels. While these methods are simple to implement, they achieve limited performance due to the challenging extraction of the shape features, which sometimes require manual measurements and graph matching techniques [116, 117].

Subspace learning methods, including Principal Components Analysis (PCA), Linear Discriminant Analysis (LDA) and force field [118], are also popular approaches to ear recognition. Chang *et al.* [119] applied PCA to both face and ear recognition and could achieve a significant improvement in performance when combining both biometrics. Hurley *et al.* [118] used force field feature extraction, which maps the ear to an energy field. The extracted features represent "potential wells" and "potential channels". More recently, Hanmandlu and Mamta [25] used the Local Principal Independent Components (LPIC) as an extension of PCA to improve the ear recognition performance. Zhang *et al.* [120] combined Independent Components Analysis (ICA) with a Radial Basis Function (RBF) to improve the performance of PCA. However, these subspace learning methods are not sufficiently resilient to image variations and thus, they perform poorly under unconstrained conditions.

Spectral approaches, which are based on extracting features from the spectral domain representation, use the local orientation information for ear recognition. Abate *et al.* [121] used a rotation invariant descriptor, the Generic Fourier Descriptor (GFD), to represent ear features. Sana *et al.* [122] used a Haar wavelet transform to represent the texture of the ear image and calculated the matching scores using the

Hamming distance. Wang *et al.* [123] used a Haar wavelet transform and Uniform Local Binary patterns (ULBPs). They decomposed the ear using the Haar wavelet transform, then they combined ULBPs with block-based and multi-resolution methods for texture feature extraction. They finally classified the features using the nearest neighbor classifier. Zhao and Mu [124] used a 2D wavelet transform to generate low frequency images, then they applied the orthogonal centroid algorithm [125] to extract the features. Kumar and Zhang [126] used log-Gabor wavelets for feature extraction and a Hamming distance for classification. Kisku *et al.* [127] used a Gaussian Mixture Model to develop an ear skin model. Tariq *et al.* [29] extracted features through Haar wavelets followed by ear identification using fast normalized cross correlation. Murukesh *et al.* [128] used a contourlet transform for feature extraction and Fisher's Linear Discriminant Analysis (FLDA) for classification. Kumar and Chan [27] used the sparse representation of local gray level orientations to efficiently recognize the ear's identity. Benzaoui *et al.* [24] showed that the binarized statistical image features (BSIF) in association with the KNN classifier yield good performance on constrained images. Jacob and Raju [26] investigated the combination of Gray Level Co-occurrence Matrix (GLCM), Local Binary Pattern (LBP) and Gabor Filter features for efficient ear recognition. Despite their popularity, spectral methods, which rely on hand-crafted features, are problem specific and cannot adapt easily to changing environments.

More recently, deep neural network (DNN) based models have achieved impressive performance in many problem domains. A DNN usually consists of layers of convolutional filters where the weights of the filters can be learned using a gradient descent based optimization procedure. This layered approach, with the addition of large amounts of training data and GPU power, was shown to yield accurate systems for classification in many application domains. AlexNet [38] was the first DNN that

achieved impressive performance on the large scale ImageNet dataset [37]. AlexNet includes techniques such as dropout for regularization, and ReLu non-linearities that still see widespread use. The VGG models [39] extend the AlexNet framework by adding more layers between pooling stages. VGG networks can be trained efficiently because all of the convolutional layers use small 3×3 filters. This also can help with over-fitting. More recently, ResNet architectures [40] build very deep networks by utilizing skip connections instead of the traditional sequential architecture. Although ResNet can be much deeper than VGG, the model size is substantially smaller due to the use of global average pooling rather than fully-connected layers.

Nevertheless, deep learning has only recently been utilized for ear recognition [30, 74, 31, 129]. One difficulty for ear recognition problems is the limited amount of labeled training data. Emersic *et al.* [30] overcame this by using data augmentation. For each training image, many similar training images were generated with slight translations, rotations, color transforms, etc. This data augmentation allowed DNNs to be fine-tuned. To further combat over-fitting caused by limited data, the work proposed selective learning, where only a subset of layers of the network were learned. AlexNet, VGG16, and SqueezeNet [74] were considered, with SqueezeNet yielding the best performance of 62% rank-1 accuracy. The authors evaluated their approach on an unconstrained ear dataset where they combined the AWE and CVLE datasets [130] in addition to 500 ear images of 50 subjects collected from the web, in order to have more data available to work with. Note that the authors did not consider the more recently introduced ResNet [40], which might achieve better performance. Galdamez *et al.* [31] built a custom neural network for recognizing ears, instead of utilizing existing pre-trained networks. The motivation for building a custom network is that it would be faster than the large pre-trained networks, however it may achieve less accuracy. Tian and Mu [131] also built a custom network with three convolutional

layers and evaluated it on the constrained USTB ear database [132]. Omara *et al.* [129] utilized pre-trained features from the VGG-m model [133] to classify the USTB constrained ear images using a pairwise SVM classifier.

Several new methods were recently presented at the Unconstrained Ear Recognition Challenge (UERC) [134]. The UERC introduced a new dataset for the challenge, based on the AWE dataset. Surprisingly, the winning entry relied on a hand crafted feature based on Chainlets [135]. Other entries attempted various methods of fine tuning or training deep networks from scratch.

4.3 Ear Recognition using Transfer Learning

Existing DNNs pre-trained on the large ImageNet dataset [37] are utilized and adapted for unconstrained ear recognition. The pre-trained feature representations provide a starting point for creating robust classifiers for unconstrained ear recognition.

4.3.1 Deep Neural Networks

DNN features from five different deep DNN architectures are explored as part of this work: AlexNet [38], VGG16 [39], VGG19 [39], ResNet18 [40], and ResNet50 [40]. Table 4.1 presents a summary of the five DNN models' characteristics. These networks have been pre-trained on the ImageNet dataset [37] that includes over 1.2 million images for 1,000 object classes.

AlexNet [38] is a DNN architecture that won the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) for image classification. The model architecture, which has 60 million parameters and 500,000 neurons, consists of five convolutional layers and three fully connected layers with a final 1000-way softmax.

VGG network architectures [39] are much deeper than AlexNet and were the

Table 4.1: Main Characteristics of Considered DNN Architectures: Design Year, Number of Parameters in Millions (Mill.), Number of Convolutional (Conv.) Layers and Number of Fully Connected (FC) Layers.

Network	Year	Parameters (Mill.)	Conv. layers	FC layers
AlexNet	2012	60	5	3
VGG16	2014	138	13	3
VGG19	2014	144	16	3
ResNet18	2015	11.7	17	1
ResNet50	2015	25.6	49	1

winner of the 2014 ILSVRC for image localization and classification. Compared with AlexNet, a single convolutional layer between pooling stages is replaced with multiple stacked convolutional layers, which are followed by three fully connected layers. The final layer is the softmax layer. The VGG style networks, which include 133 million to 144 million parameters, use small 3×3 size filters to reduce the number of parameters and consequently reduce over-fitting. In this work, the VGG16 and VGG19 architectures are used, where 16 and 19 refer to the number of trainable layers.

ResNet [40], which won the 2015 ILSVRC, made the concept of training very deep neural networks possible and less challenging. The network uses "skip" connections between convolutional blocks in order to create much deeper neural networks. The skip connections ensure that there is no vanishing gradient problem. The layers are formulated as learning residual functions with respect to the layer inputs, instead of learning more simple feed-forward functions. Despite of their large depth, ResNets have much less number of parameters varying between 11.7 million (18 layers) and 60.2 million (152 layers). In this work, the 18-layer ResNet18 and 50-layer ResNet50 models are used.

4.3.2 Extracting Deep Features

Features extracted from DNNs have been shown to achieve good performance on many different problem domains [32]. When a network is trained on a large diverse dataset such as ImageNet, features extracted from network layers can be transferred to other problems, in this case ear recognition.

Similar to [129], a linear SVM is trained using features extracted from the DNNs described in Section 4.3.1. However, different from [129], the unconstrained ear recognition problem is considered in this work. Since this is more difficult than the constrained problem addressed in [129], data augmentation techniques are incorporated to improve the accuracy. Additionally, the performance is evaluated using five different network architectures, where it is shown that the choice of architecture can significantly affect the resulting classification accuracy. Furthermore, features extracted from different layers of the same network can give different classification accuracies. An exhaustive search on the layers is performed and results with the layer that gives the highest accuracy are reported using the AWE and CVLE datasets [130]. It is found that the best performance corresponds to the last convolutional layer for AlexNet and VGG16, the second to last convolutional layer for VGG19, and the last convolutional layer of the third residual block for the ResNets. The LibSVM library [136] is used to train a one-against-one multi-class linear SVM using the extracted features. The very high dimensionality of the extracted features makes SVM training computationally expensive, so Principal Component Analysis (PCA) is used to reduce the dimensionality of the features while retaining 99% of the feature variance.

4.3.3 Fine Tuning

The work in this section has been performed by the co-authors in [11]. For completeness of the proposed unconstrained ear recognition method, this work is described next.

While the features from the fixed pre-trained networks can be useful for ear recognition, a more accurate classifier can be trained by *fine-tuning* the parameters of the neural network. Fine-tuning is essentially training the network for several more iterations on a new dataset. This process will adapt the generic filters trained on the ImageNet dataset to the ear recognition problem.

The same networks described in Section 4.3.2 are used. For each network, the last fully connected layer is replaced with a new fully connected layer where the number of units is equal to the number of classes in the dataset. The parameters of the new fully connected layer are initialized by Glorot initialization [137]. The network is trained for 25 epochs using stochastic gradient descent. At around 25 epochs, all of the network architectures achieve near 100% accuracy on the training set, so no more improvement in training can be achieved. The learning rate of the last layer is set to 0.1 and the learning rate of all of the other pre-trained layers are set to 0.01. This is because the last layer is trained from scratch whereas the other layers are initialized with pre-trained weights.

This fine-tuning approach is different from [30]. The method of [30] performs "selective" learning where the early layers are fixed and later layers are fine-tuned. This approach allows the early layers to adapt, but at a smaller learning rate than the last layer. This is also different than the "full training" of [30] because the learning rates of different layers are not all the same.

The networks are fine-tuned using data augmentations as explained in Section

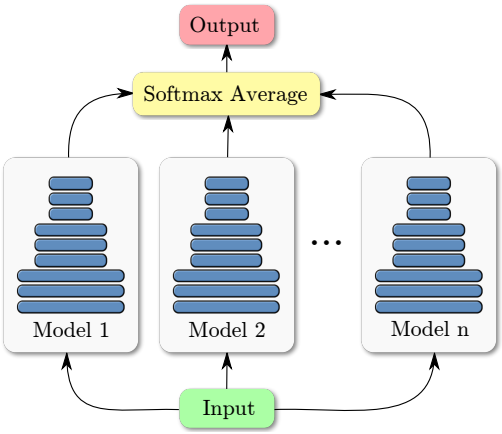


Figure 4.1: Structure of ensemble models [11]. The ensemble consists of n models. The parameters of the last fully connected layer of each model are initialized with different random values and each model is trained separately. During testing, the soft-max outputs of the constituent models are averaged to yield the final output prediction.

4.4.3. However, even with data augmentation, the fine-tuned deep networks may over-fit the new training data. This is particularly a problem in ear recognition because the datasets are relatively small. An averaging ensemble is used, in addition to data augmentation, to reduce the effect of over-fitting. Ensembles of five models are tested, where the last layer of each ensemble member is initialized with different random values. The different initializations yield different local minima after the network has been trained. To obtain a final output prediction during testing, the average of the soft-max outputs of the ensemble members is taken. The final predicted label is the argmax of the averaged soft-max outputs. The full ensemble model can be seen in Figure 4.1.

4.4 Experimental Setup and Results

In the experiments, the results are reported for all five CNN architectures. Results for deep feature extraction and fine tuning are presented, as described in Section 4.3. The results with and without data augmentation are compared for deep feature

extraction, but for all of the fine-tuning experiments, the augmented datasets are used.

4.4.1 Datasets

The experiments are performed on two publicly available unconstrained ear datasets: AWE and CVLE [130]. Both datasets consist of images captured "in the wild" of the ears of public figures collected by a web crawler. The images include realistic variations, such as contrast/illumination, occlusion, head rotations, gender, race, visual quality distortions and image resolution. These datasets are considered challenging for automatic ear recognition applications.

The AWE dataset includes 1000 images of 100 persons (10 images/person), while the CVLE dataset includes 804 images for 16 persons (on average 50.25 images/person). For both datasets, the images come in different sizes varying from 15×29 pixels to 473×1022 pixels. All images are tightly cropped and do not include the face. Figure 4.3 and Figure 4.4 show sample images from both datasets.

Additionally, the AWE and CVLE datasets are combined to form a third dataset (AWE+CVLE). For this dataset, the same train and test splits are used as in the respective datasets.

4.4.2 Experimental Protocols

The given training/testing split provided in the AWE toolbox [130] is used. The training set consists of 60% of the images and the testing set includes the remaining 40%. For the CVLE dataset, the dataset is randomly split into 60% training images and 40% testing images.

As in [30], identification experiments are performed with a closed-set experimental protocol, where the proposed models should predict the class to which the input image

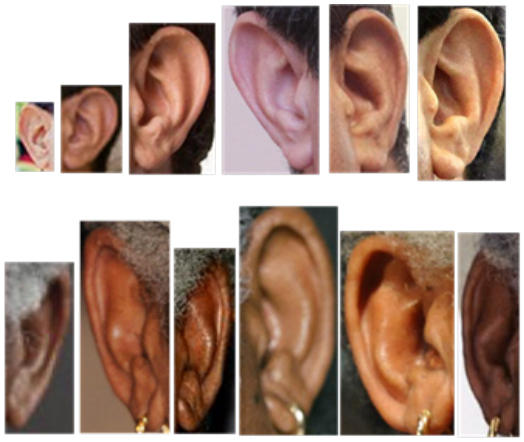


Figure 4.2: Sample images from the AWE dataset. Each row corresponds to the images of one subject. The images include variations of head rotation, illumination, gender, race, occlusion, blurring and image resolution.

Figure 4.3: Sample images from the CVLE dataset. Each row corresponds to the images of one subject. The images include variations of head rotation, illumination, gender, occlusion and image resolution.

belongs. There are 100 classes for AWE, 16 classes for CVLE and 116 classes for the combined dataset (AWE+CVLE). For performance evaluation, rank-1 and rank-5 recognition rates, as well as Cumulative Match-score Curves (CMC) are used. The Cumulative Match-score Curve is formed by computing the recognition rate using the top i predictions from the model, where i varies from 1 to m, and m is the number of classes.

For the single fine-tuned models, average performance is reported over five random seeds. These five models are the same models used in the averaging ensemble.

All of the neural networks operate on a $224 \times 224 \times 3$ size input. Thus, the orig-

inal images are resized to this dimension before feeding them to the neural network. Both bilinear and bicubic interpolation are considered for resizing the images. Both methods were found to produce similar results in terms of classification performance. The mean of the ImageNet dataset is additionally subtracted from the input images.

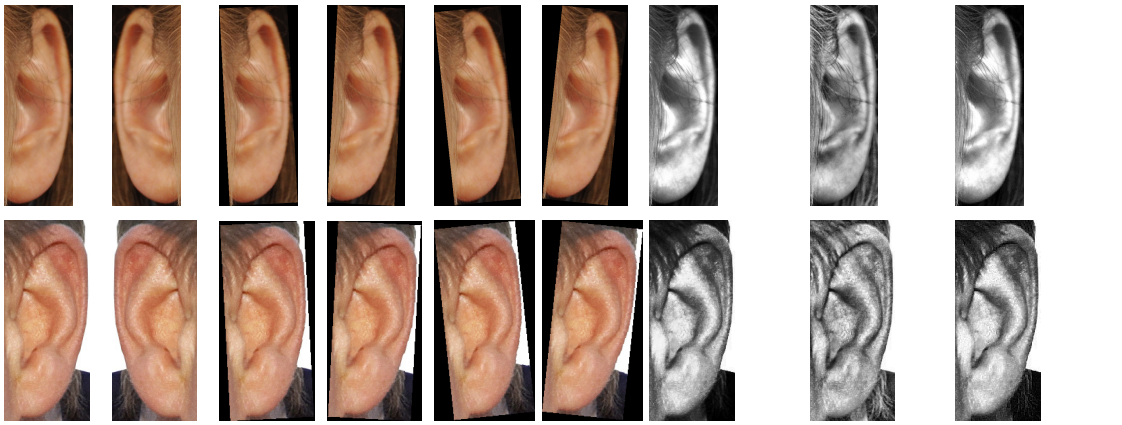


Table 4.2: Data Augmentation Examples. Each Row Corresponds to a Single Source Image of One Subject. The Third to Sixth Images Include Rotated Images with Angles +3, -3, +6, and -6 Degrees using Nearest Neighbor Interpolation. The Remaining Images Include the Other Four Augmentation Variations in Addition to the Original Image.

4.4.3 Data Augmentation

The datasets are relatively small, so it is easy for models to over-fit and not generalize well on testing data. To alleviate this problem, the dataset is augmented by a factor of 9 with several image transformations as shown in Figure 4.2. Image transformations are selected to introduce spatial as well as pixel value variations. Noting that the head rotation in the source images is not constant, augmentation is considered with moderate rotations using nearest neighbor interpolation (-6, -3, +3,

Table 4.3: Rank-1 and Rank-5 Accuracy (%) of Models Trained and Tested on the AWE Dataset. Bold Entries Denote Best Performers and Underlined Entries Denote Second Best Performers.

	Rank-1				Rank-5			
	Deep Features	Deep Features (augmentation)	Fine-tune Single	Fine-tune Ensemble	Deep Features	Deep Features (augmentation)	Fine-tune Single	Fine-tune Ensemble
AlexNet	34.25	46.75	37.50	45.00	55.50	73.50	62.70	71.00
VGG16	31.25	49.25	50.70	<u>66.00</u>	53.75	70.25	74.65	81.50
VGG19	40.25	56.25	50.25	65.75	64.25	76.75	74.45	<u>84.75</u>
ResNet18	31.75	61.50	56.35	**68.50**	57.50	**85.00**	74.80	83.00
ResNet50	40.75	63.00	48.40	56.25	66.50	80.25	70.65	77.50

+6 degrees) to increase the classifier's robustness to rotation. Although the left and right ears are not necessarily the same, horizontal flipping is applied to expose the classifier to more variations of ear structures. Next, three normalization techniques are considered to introduce pixel value variations: histogram equalization, adaptive histogram equalization, and wavelet-based normalization. These three latter transformations are applied to grayscale versions of the source image. Grayscale images force the classifier to use texture information, rather than rely on color information. The histogram equalization methods spread image intensities in the spatial domain, while the wavelet-based illumination method enhances the contrast in the wavelet domain. For each grayscale image, the grayscale channel is replicated such that the resulting image is of size $224 \times 224 \times 3$. This is done so that both the grayscale images and the color images can be fed to the networks.

4.4.4 Feature Extraction Results

In the first series of experiments, the performance of the five DNN architectures is evaluated by assessing the representation ability of their respective deep features. Experiments are constructed as described in Section 4.3.2. For these experiments,

Table 4.4: Rank-1 and Rank-5 Accuracy (%) of Models Trained and Tested on the CVLE Dataset. Bold Entries Denote Best Performers and Underlined Entries Denote Second Best Performers.

	Rank-1				Rank-5			
	Deep Features	Deep Features (augmentation)	Fine-tune Single	Fine-tune Ensemble	Deep Features	Deep Features (augmentation)	Fine-tune Single	Fine-tune Ensemble
AlexNet	77.57	79.13	85.86	89.10	96.57	98.13	97.76	97.82
VGG16	79.13	81.93	90.16	93.15	95.02	97.82	99.37	<u>99.38</u>
VGG19	86.29	86.60	89.41	92.52	96.57	98.75	98.07	**99.69**
ResNet18	87.54	<u>93.46</u>	90.59	<u>93.46</u>	93.46	<u>99.38</u>	99.19	<u>99.38</u>
ResNet50	86.92	92.83	91.40	**94.08**	97.51	99.03	98.87	**99.69**

the effect of data augmentation on the deep networks' performance is analyzed.

Tables 4.3, 4.4, and 4.5 show the rank-1 and rank-5 accuracies for the feature extraction-based models trained with original and augmented versions of the AWE, CVLE, and combined AWE+CVLE datasets, respectively. Overall, the network features perform better on CVLE compared with AWE due to the lower number of classes and larger number of training images per class for the CVLE dataset. Data augmentation is able to improve the accuracies by an average of 30% for AWE, 3% for CVLE, and 30% for their combination. The impact of data augmentation is larger on the AWE dataset than on the CVLE dataset due to the lack of sufficient training samples per class for the AWE dataset. Fig. 4.4 shows the CMC curves for the ResNet18 features performance on the three considered datasets. ResNet18 features show better performance when the training set is augmented for all considered datasets. This trend is also seen for all the other considered networks' features.

ResNet features consistently perform the highest across the three datasets. For the augmented AWE (Table 4.3) and combined (Table 4.5) datasets, ResNet50 features achieve the highest rank-1 accuracies of respectively 63% and 65.6%. For the augmented CVLE (Table 3), ResNet18 features achieve the highest rank-1 accuracy

Table 4.5: Rank-1 and Rank-5 Accuracy (%) of Models Trained and Tested on the Combined AWE + CVLE Dataset. Bold Entries Denote Best Performers and Underlined Entries Denote Second Best Performers.

	Rank-1				Rank-5			
	Deep Features	Deep Features (augmented)	Fine-tune Single	Fine-tune Ensemble	Deep Features	Deep Features (augmented)	Fine-tune Single	Fine-tune Ensemble
AlexNet	41.89	55.76	58.39	66.16	66.71	79.47	79.53	84.74
VGG16	43.00	54.37	68.99	77.95	64.91	78.64	86.29	90.29
VGG19	49.38	64.49	68.90	<u>78.92</u>	73.51	82.39	86.43	90.29
ResNet18	46.19	64.91	71.87	**80.03**	69.35	84.33	86.68	**93.48**
ResNet50	58.11	65.60	69.90	75.73	76.84	84.88	85.55	<u>90.85</u>

Table 4.6: Rank-1 and Rank-5 Accuracy (%) of Models Trained on the Combined AWE+CVLE Dataset, and Tested only on the Images from the AWE Dataset. Bold Entries Denote Best Performers and Underlined Entries Denote Second Best Performers.

	Rank-1				Rank-5			
	Deep Features	Deep Features (augmented)	Fine-tune Single	Fine-tune Ensemble	Deep Features	Deep Features (augmented)	Fine-tune Single	Fine-tune Ensemble
AlexNet	15.75	35.75	42.75	51.75	45.50	67.25	70.00	77.50
VGG16	14.25	30.75	56.85	68.00	42.75	64.75	80.00	85.00
VGG19	21.75	45.25	55.60	<u>68.75</u>	57.00	72.75	79.25	84.75
ResNet18	44.50	44.75	57.25	**69.25**	67.25	73.25	78.95	**88.75**
ResNet50	36.00	45.00	54.75	63.00	60.50	73.50	77.75	<u>85.50</u>

of 93.46%.

The scenario where the training set is the combined AWE+CVLE dataset is also tested, but the test set is restricted to only the AWE dataset. The results are presented in Table 4.6. It can be seen from Table 4.6 that the performance of all feature extraction-based models decreases by an average of 20% as compared to using the AWE training set (Table 4.3). This shows that the SVM is unable to utilize the extra

data to increase classification accuracy. In fact, the presence of extra classes introduces more avenues for error. Since the LibSVM uses a one-against-one technique for multi-class classification, binary SVMs trained with AWE class pairs remain the same, but the addition of binary SVMs trained with AWE-CVLE class pairs can introduce error. Additionally, the CVLE dataset has more training images per class, which may bias the classifier to erroneously predict CVLE classes when AWE classes should be predicted.

4.4.5 Fine-Tuning Results

In the second series of experiments, the previous results are compared with the results of the fine-tuning procedures on all five DNN networks. For these results, augmented training sets are only used. The results (Tables 4.3 to 4.5) show that the ensemble method significantly outperforms the single model performance by an average of 20% for AWE, 3% for CVLE, and 12% for the combined AWE+CVLE dataset. The CMC curves show the difference in performance between a single model and the five-member ensemble for the ResNet18 model (Figure 4.5). Among the five model architectures, ResNet18 achieves the best performance for the AWE and combined (AWE+CVLE) datasets with respectively rank-1 recognition accuracies of 58.35% and 71.87% for the single model, 68.5% and 80.03% for the ensemble model. ResNet50 performs the best for the CVLE dataset with rank-1 recognition accuracies of 91.85% and 94.08% for the single and ensemble models, respectively.

Surprisingly, the fine-tuned DNNs do not always give better performance than the feature extraction-based models. For example, on the AWE dataset, the rank-1 accuracy of each feature extraction-based model is nearly always better than the corresponding single fine-tuned model (except for the VGG16 model). For example, the ResNet18 feature-based model achieves 61.50% rank-1 accuracy while the

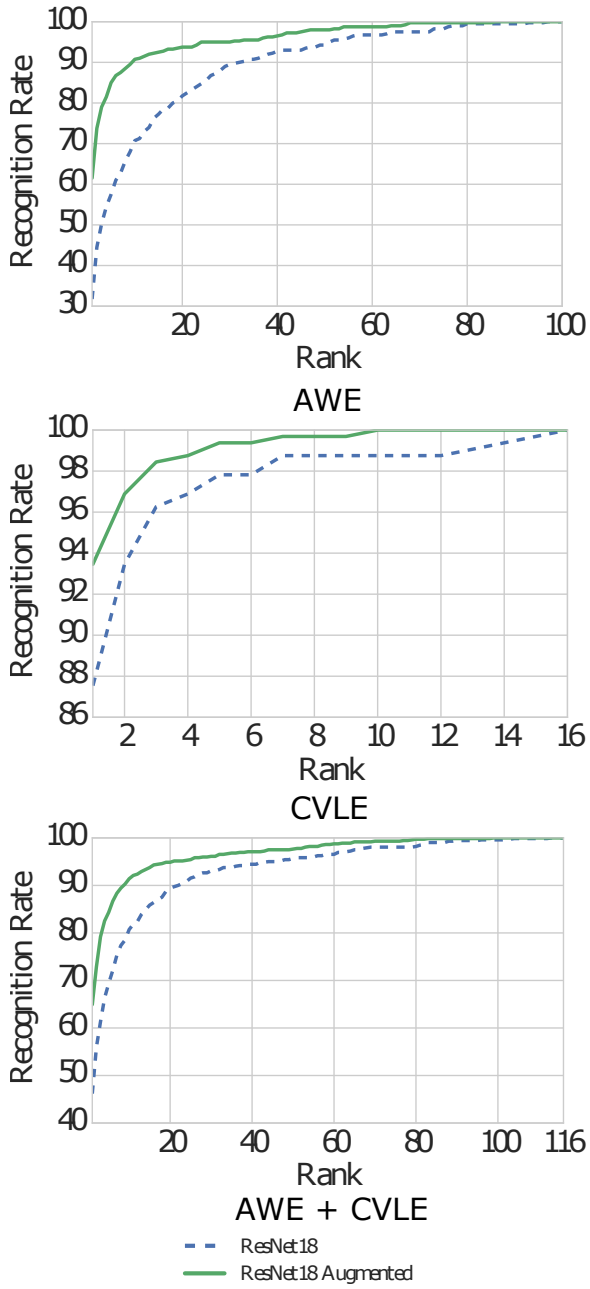


Figure 4.4: CMC curves for ResNet-18 feature-based SVM models. The models perform better for all ranks when the training data is augmented.

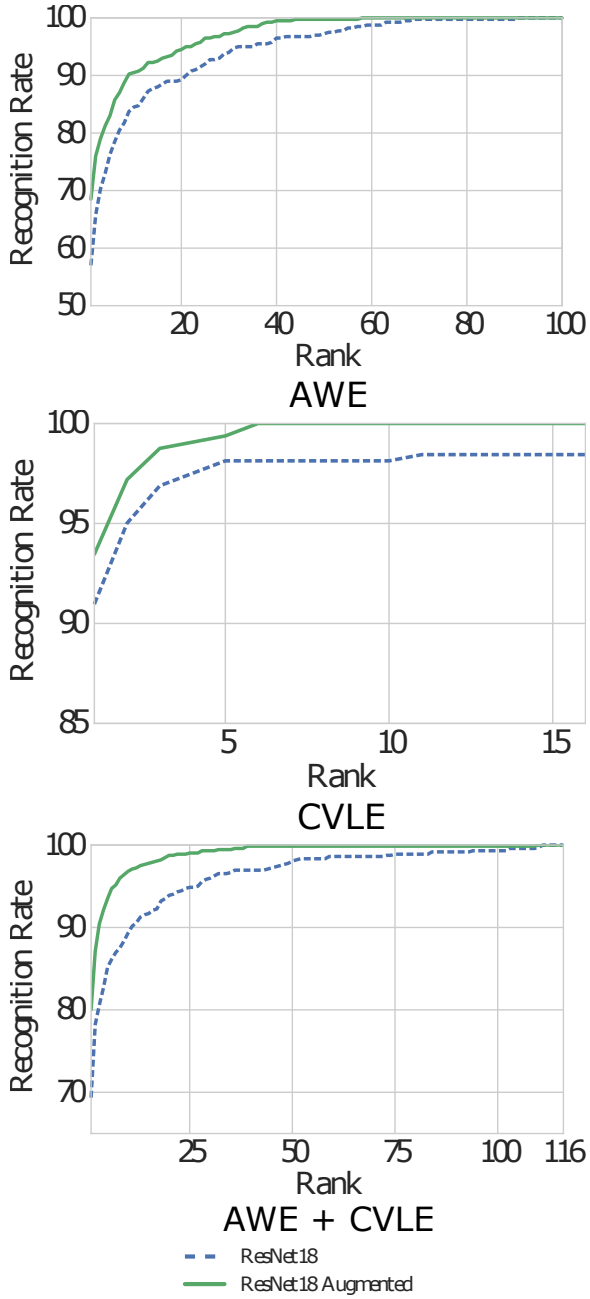


Figure 4.5: CMC curves for the fine-tuned single ResNet18 model and five member ensemble ResNet18 model. The ensemble model performs better than the single ResNet18 model for all ranks.

single fine-tuned model achieves only a 56.35% rank-1 accuracy. However, for the combined dataset, the fine-tuned networks always outperform the corresponding feature extraction-based model. For the combined dataset, the ResNet18 feature-based model gives a much lower 44.75% rank-1 accuracy, while the ResNet18 single fine-tuned model gives a 57.25% rank-1 accuracy. The five member ensemble model gives a much higher rank-1 accuracy of 69.25%. These results show that the single fine-tuned models perform best when there is a larger amount of data. For small datasets, feature extraction-based models may be more appropriate as compared to single fine-tuned models, because they are not as susceptible to over-fitting. The ensemble DNN model outperform both feature-based and single fine-tuned models in almost all cases even for small datasets.

For the combined dataset, the trained fine-tuned single and ensemble models are separately evaluated on the AWE dataset (Table 4.6). The network trained on the combined datasets has the advantage of having more data to potentially learn better feature representations. Compared with the results in Table 4.3, the networks yield higher accuracy, despite the increased number of classes in the training set. This is in contrast to the feature extraction-based models that could not use the increased data to learn better feature representations.

4.5 Conclusion and Future Work

A method for utilizing DNNs for unconstrained ear recognition is proposed. Five state-of-the-art DNNs are utilized, however, the methods presented in this work could be used with newer DNN architectures to achieve better performance. It is shown that, in the case where long training time is not desirable or a large amount of data is not available, the features from pre-trained DNNs can be used with a shallow classifier to give a comparable performance to fine-tuned networks. If more accuracy is desired

and a relatively large amount of training data is available, the pre-trained networks can be fine-tuned by training on the ear datasets. Further increases in classification accuracy is achieved, independent of the size of the datasets, by creating an ensemble of fine-tuned networks. Overall, the best results are achieved with an ensemble of ResNet18 models, which provides consistent performance across the tested datasets. This indicates that the residual connections used in ResNets are useful for the ear recognition. On average, the ResNet18 ensemble outperforms the ResNet50 ensemble because of the ResNet18 model fewer parameters, which makes it less susceptible to overfitting.

Chapter 5

CONCLUSION

This dissertation presents robust methods for biometric (face/ear) images under unconstrained conditions. The work mainly focuses on three main quality distortions, which are blur, occlusion and noise. This chapter summarizes the main contributions of this work and suggests possible future research directions.

5.1 Contributions

The main contributions of this work are as follows:

- Based on the popular SRC, an Augmented SRC-based framework (ASRC) that is more robust to blur, occlusion and noise for a selected feature is proposed. The proposed model accounts for the blur/occlusion distortion as part of the dictionary construction. The ASRC is a novel sparse-based framework for face recognition in the presence of visual quality distortions.

- While SRC can handle the additive occlusion/corruption covering less than 33 percent of the image size, the occlusion distortion is represented in this work as a linear pointwise multiplicative operation instead of an additive one and is used as part of the dictionary construction in the proposed ASRC framework. The proposed ASRC results in a significantly higher classification performance, as compared to SRC, and is shown to be able to handle block occlusions much larger than 33 percent of the image size.

- The importance of feature selection is explored under various levels of blur, occlusion and noise for the SRC classifier. A Feature Sparse Coding and Classi-

fication Index (FSCCI) is proposed as a metric that measures the sparse coding ability and the classification performance of features used within the SRC and the proposed ASRC frameworks. The FSCCI can be used for feature selection in order to maximize the classification performance of SRC and ASRC.

- A new LFW dataset is constructed for face identification including five subsets of images: clean images, Gaussian blur distorted images (4 levels of distortion), realistic blur distorted images (8 levels of distortion), single/double occluded distorted images (3 levels of distortion) and white noise distorted images (4 levels of distortion). The constructed LFW face identification dataset will be made available at ivulab.asu.edu.

- A solution to unconstrained ear recognition is proposed by transfer learning based on features from pre-trained DNNs.

5.2 Future Work

There are several directions that can be explored in future work:

- The ASRC framework can be extended to solve other uncontrolled image variations issues, such as extreme pose changes. A possible direction would be to augment the dictionary this time with different pose variations at different angles and consider each level as a separate object class. This remains an interesting direction for future work, as this type of variations is not well handled by the SRC model.

- Similar to face recognition, the unconstrained ear recognition can be investigated in the presence of high levels of noise, blur and occlusion on the AWE and CVLE unconstrained ear datasets and the proposed ASRC method can be evaluated on the distorted ear datasets.

- As deep neural networks are not resilient to quality distortions, distortion-robust DNN architectures can be designed to mitigate the effects of high levels of blur, occlusion and additive noise.
- Recognition methods for misaligned unconstrained face/ear images can be improved by designing an alignment method that works on distorted images.

REFERENCES

[1] Freeman W., Durand F., Weiss Y., and Levin A. Understanding and evaluating blind deconvolution algorithms. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2009.

[2] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.

[3] 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, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[4] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[5] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[6] Z. H. Zhou and X. Geng. Projection functions for eye detection. *Pattern Recognition*, 37(5):1049–1056, 2004.

[7] A. V. Iannarelli. *Ear identification.* Paramont Publishing Company, 1989.

[8] O.M. Parkhi, A.Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, pages 1–12, 2015.

[9] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[10] T. Hassner, S. Harel, E. Paz, and R. Enbar. Effective face frontalization in unconstrained images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4295–4304, 2015.

[11] J. Mounsef S. Dodge and L. Karam. Unconstrained ear recognition using deep neural networks. *IET Biometrics*, 2018.

[12] J. Edgell and A. Trimpe. Limitations of facial recognition technology. http://www.fedtechmagazine.com/article/2013/11/4-limitations-facial-recognition-technology, Nov. 22, 2013 (accessed Oct. 8, 2017).

[13] G. Hua, M.-H. Yang, E. Learned-Miller, M. Turk Y. Ma, D. J. Kriegman, and T. S. Huang. Introduction to the special section on real-world face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):1921–1924, 2011.

[14] S. Shan, W. Gao, B. Cao, and D. Zhao. Illumination normalization for robust face recognition against varying lighting conditions. In *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, pages 157–164, 2003.

[15] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing*, 19(6):1635–1650, 2010.

[16] J. Holappa, T. Ahonen, and M. Pietikinen. An optimized illumination normalization method for face recognition. In *IEEE International Conference on Biometrics: Theory, Applications and Systems*, pages 1–6, 2008.

[17] G. Aggarwal, S. Biswas, and R. Chellappa. UMD experiments with FRGC data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 172–172, 2005.

[18] M. Yang, L. Zhang, J. Yang, and D. Zhang. Robust sparse coding for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 625–632, 2011.

[19] J. Pillai, V. M. Patel, R. Chellappa, and N. K. Ratha. Towards a practical face recognition system: Robust registration and illumination by sparse representation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1838–1841, 2010.

[20] C.-P. Wei, C.-F Chen, and Y.-C. F. Wang. Robust face recognition with structurally incoherent low-rank matrix decomposition. *IEEE Transactions on Image Processing*, 23(8):3294–3307, 2014.

[21] S. Liao, A. K. Jain, and S. Z. Li. Partial face recognition: Alignment-free approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(5):1193–1205, 2013.

[22] W. Deng, J. Hu, and J. Guo. Extended SRC: Undersampled face recognition via intraclass variant dictionary. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1864–1870, 2012.

[23] J. Wright, A. Y. Yang, Y. Allen, A. Ganesh, S. Sastry, S. Shankar, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.

[24] A. Benzaoui, I. Adjabi, and A. Boukrouche. Person identification based on ear morphology. In *International Conference on Advanced Aspects of Software Engineering*, pages 1–5, Oct 2016.

[25] M. Hanmandlu and Mamta. Robust ear based authentication using local principal independent components. *Expert Systems with Applications*, 40(16):6478–6490, 2013.

[26] L. Jacob and G. Raju. Ear recognition using texture features-a novel approach. In *Advances in Signal Processing and Intelligent Recognition Systems*, pages 1–12, 2014.

[27] A. Kumar and T. S. Chan. Robust ear identification using sparse representation of local texture descriptors. *Pattern Recognition*, 46(1):73–85, 2013.

[28] A. S. Anwar, K. K. Ghany, and H. Elmahdy. Human ear recognition using geometrical features extraction. *Procedia Computer Science*, 65:529–537, 2015.

[29] A. Tariq, M. A. Anjum, and M. U. Akram. Personal identification using computerized human ear recognition system. In *International Conference on Computer Science and Network Technology*, volume 1, pages 50–54, Dec 2011.

[30] Z. Emersic, D. Stepec, V. Struc, and P. Peer. Training convolutional neural networks with limited training data for ear recognition in the wild. In *Automatic Face Gesture Recognition*, pages 987–994, May 2017.

[31] P.L. Galdamez, L. Pedro, W. Raveane, and A. G. Arrieta. A brief review of the ear recognition process using deep neural networks. *Journal of Applied Logic*, 24:62–70, 2016.

[32] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813, 2014.

[33] F. S. Samaria and A. C. Harter. Parameterisation of a stochastic model for human face identification. In *IEEE Workshop on Applications of Computer Vision*, pages 138–142, 1994.

[34] Z. Zhang, E. Klassen, and A. Srivastava. Gaussian blurring-invariant comparison of signals and images. *IEEE Transactions on Image Processing*, 22(8):3145–3157, 2013.

[35] J. Flusser, S. Farokhi, C. Höschl, T. Suk, B. Zitová, and M. Pedone. Recognition of images degraded by Gaussian blur. *IEEE Transactions on Image Processing*, 25(2):790–806, 2016.

[36] P. Vageeswaran, K. Mitra, and R. Chellapa. Blur and illumination robust face recognition via set-theoretic characterization. *IEEE Transactions on Image Processing*, 22(4):1362–1372, 2013.

[37] O. Russakovsky, J. Deng, H. Su, J.Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[38] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2014.

[40] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[41] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[42] A. J. Smola, O. L. Mangasarian, and B. Schölkopf. Sparse kernel feature analysis. In *Classification, Automation, and New Media*, pages 167–178. Springer, 2002.

[43] S. Mika, J. Weston G. Ratsch, B. Scholkopf, and K. R. Mullers. Fisher discriminant analysis with kernels. In *Proceedings of the IEEE Signal Processing Society Workshop.*, pages 41–48, 1999.

[44] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, 1962.

[45] J. Flusser and T. Suk. Degraded image analysis: an invariant approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):590–603, 1998.

[46] J. Flusser and B. Zitová. Combined invariants to linear filtering and rotation. *International Journal of Pattern Recognition and Artificial Intelligence*, 13(08):1123–1135, 1999.

[47] J. Flusser, T. Suk, and S. Saic. Recognition of blurred images by the method of moments. *IEEE Transactions on Image Processing*, 5(3):533–538, 1996.

[48] H. Zhang, H. Shu, G. N. Han, G. Coatrieux, L. Luo, and J. L. Coatrieux. Blurred image recognition by legendre moment invariants. *IEEE Transactions on Image Processing*, 19(3):596–611, 2010.

[49] Z. Zhang, E. Klassen, A. Srivastava, P. Turaga, and R. Chellappa. Blurring-invariant riemannian metrics for comparing signals and images. In *IEEE International Conference on Computer Vision*, pages 1770–1775, 2011.

[50] J. Flusser and T. Suk. Pattern recognition by affine moment invariants. *Pattern Recognition*, 26(1):167–174, 1993.

[51] T. H. Reiss. The revised fundamental theorem of moment invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):830–834, 1991.

[52] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.

[53] B. B. Manjunath, R. Chellappa, and C. von der Malsburg. A feature based approach to face recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 373–378, 1992.

[54] L. Wiskott, N. Krüger, N. Kuiger, and C. Von Der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–779, 1997.

[55] A. Lanitis, C. J. Taylor, and T. F. Cootes. Automatic face identification system using flexible appearance models. *Image and Vision Computing*, 13(5):393–401, 1995.

[56] B. Duc, S. Fischer, and J. Bigun. Face authentication with gabor information on deformable graphs. *IEEE Transactions on Image Processing*, 8(4):504–516, 1999.

[57] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[58] M. Choras. Ear biometrics based on geometrical method of feature extraction. pages 51–61, 2004.

[59] M. Choras and R. S. Choras. Geometrical algorithms of ear contour shape representation and feature extraction. In *International Conference on Intelligent Systems Design and Applications*, volume 2, pages 451–456, Oct 2006.

[60] G. C. Feng and P. C. Yuen. Variance projection function and its application to eye detection for human face recognition. *Pattern Recognition Letters*, 19(9):899–906, 1998.

[61] G. C. Feng and P. C. Yuen. Multi-cues eye detection on gray intensity image. *Pattern Recognition*, 34(5):1033–1046, 2001.

[62] V. Ngoc Son. Titre: Contributions à la reconnaissance de visages à partir d'une seule image et dans un contexte non-contrôlé. *Ph.D. Thesis*, 2010.

[63] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[64] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In *European Conference on Computer Vision*, pages 469–481, 2004.

[65] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.

[66] V. Ojansivu and J. Heikkila. Blur insensitive texture classification using local phase quantization. In *International Conference on Image and Signal Processing*, pages 236–243, 2008.

[67] J. Kannala and E. Rahtu. Bsif: Binarized statistical image features. In *International Conference on Pattern Recognition*, pages 1363–1366, 2012.

[68] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006.

[69] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[70] W. Liu, Y. Wen, Z. Yu, B. Raj M. Li, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, 2017.

[71] S. Dodge and L. Karam. Understanding how image quality affects deep neural networks. In *International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6, 2016.

[72] K. Grm, V. Struc, A. Artiges, M. Caron, and H. Ekenel. Strengths and weaknesses of deep learning models for face recognition against image degradations. *IET Biometrics*, 7(1):81–89, 2017.

[73] C. Szegedy, W. Liu, Y. Jia, Sermanet P, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[74] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and ¡ 0.5 MB model size. *arXiv preprint arXiv:1602.07360*, 2016.

[75] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[76] David L Donoho. Neighborly polytopes and sparse solutions of underdetermined linear equations. 2005.

[77] X.Luand and X.Li. Group sparse reconstruction for image segmentation. *Neurocomputing*, 136:41–48, 2014.

[78] M. Elad, M. A. T. Figueiredo, and Y. Ma. On the role of sparse and redundant representations in image processing. *Proceedings of the IEEE*, 98(6):972–982, 2010.

[79] A. N. Akansu and R. A. Haddad. *Multiresolution signal decomposition: transforms, subbands, and wavelets*. Academic Press, 2001.

[80] J.-L. Starck, F. Murtagh, and J. M. Fadili. *Sparse image and signal processing: wavelets, curvelets, morphological diversity*. Cambridge University Press, 2010.

[81] M. Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer-Verlag, 2010.

[82] A.M.Bruckstein, D.L.Donoho, and M.Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM review*, 51(1):34–81, 2009.

[83] Y. Xu, D. Zhang, J. Yang, and J.-Y. Yang. A two-phase test sample sparse representation method for use with face recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(9):1255–1262, 2011.

[84] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan. Sparse representation for computer vision and pattern recognition. *Proceedings of the IEEE*, 98(6):1031–1044, 2010.

[85] Y. Zhang, C. Wen, Y. Zhang, and Y. C. Soh. Determination of blur and affine combined invariants by normalization. *Pattern Recognition*, 35(1):211–221, 2002.

[86] T. Suk and J. Flusser. Combined blur and affine moment invariants and their use in pattern recognition. *Pattern Recognition*, 36(12):2895–2907, 2003.

[87] C. Y. Wee and R. Paramesran. Derivation of blur-invariant features using orthogonal legendre moments. *IET Computer Vision*, 1(2):66–77, 2007.

[88] X. Dai, H. Zhang, H. Shu, and L. Luo. Image recognition by combined invariants of legendre moment. In *IEEE International Conference on Information and Automation*, pages 1793–1798, 2010.

[89] X. Dai, H. Zhang, H. Shu, L. Luo, and T. Liu. Blurred image registration by combined invariant of legendre moment and harris-laplace detector. In *Fourth Pacific-Rim Symposium on Image and Video Technology*, pages 300–305, 2010.

[90] H. Zhu, Min M. Liu, H. Ji, and Y. Li. Combined invariants to blur and rotation using zernike moment descriptors. *Pattern Analysis and Applications*, 13(3):309–319, 2010.

[91] B. Chen, H. Shu, H. Zhang, G. Coatrieux, L. Luo, and J. L. Coatrieux. Combined invariants to similarity transformation and to blur using orthogonal zernike moments. *IEEE Transactions on Image Processing*, 20(2):345–360, 2011.

[92] H. Ji and H. Zhu. Degraded image analysis using zernike moment invariants. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1941–1944, 2009.

[93] Q. Li, H. Zhu, and Q. Liu. Image recognition by combined affine and blur tchebichef moment invariants. In *International Congress on Image and Signal Processing*, volume 3, pages 1517–1521, 2011.

[94] V. Ojansivu and J. Heikkila. Image registration using blur-invariant phase correlation. *IEEE signal Processing Letters*, 14(7):449–452, 2007.

[95] Ville Ojansivu and Janne Heikkilä. A method for blur and affine invariant object recognition using phase-only bispectrum. In *International Conference Image Analysis and Recognition*, pages 527–536, 2008.

[96] S. Tang, Y. Wang, and Y. W. Chen. Blur invariant phase correlation in x-ray digital subtraction angiography. In *IEEE/ICME International Conference on Complex Medical Engineering*, pages 1715–1719, 2007.

[97] M. Pedone, J. Flusser, and J. Heikkilä. Blur invariant translational image registration for n-fold symmetric blurs. *IEEE Transactions on Image Processing*, 22(9):3676–3689, 2013.

[98] M. Pedone, J. Flusser, and J. Heikkilä. Registration of images with n-fold dihedral blur. *IEEE Transactions on Image Processing*, 24(3):1036–1045, 2015.

[99] B. Xiao, J. F. Ma, and J. T. Cui. Combined blur, translation, scale and rotation invariant image recognition by radon and pseudo-fourier–mellin transforms. *Pattern Recognition*, 45(1):314–321, 2012.

[100] R. Gopalan, S. Taheri, P. Turaga, and R. Chellappa. A blur-robust descriptor with applications to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6):1220–1226, 2012.

[101] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.

[102] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1891–1898, 2014.

[103] Enrique Fernandez. Performance analysis of deep neural networks on objects with occlusions. Technical report, Massachussets Institute of Technology, Dec 2016.

[104] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. *Computer Vision*, pages 469–481, 2004.

[105] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher vector faces in the wild. In *BMVC*, volume 2, page 4, 2013.

[106] T. Zhu and L. Karam. A no-reference objective image quality metric based on perceptually weighted local noise. *EURASIP Journal on Image and Video Processing*, 2014(1):5, 2014.

[107] C. Q. Mylene M. Farias and K. Sanjit S. K. Mitra. No-reference video quality metric based on artifact measurements. In *IEEE International Conference on Image Processing*, volume 3, pages III–141. IEEE, 2005.

[108] M. G. Choi, J. H. Jung, and J. W. Jeon. No-reference image quality assessment using blur and noise. *International Journal of Computer Science and Engineering*, 3(2):76–80, 2009.

[109] X. He, S. Yan, Y. Hu, P. Niyogi, and H. J. Zhang. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):328–340, 2005.

[110] K. C. Lee, J. Ho, and D. J. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):684–698, 2005.

[111] A. Albiol, D. Monzo, A. Martin, J. Sastre, and A. Albiol. Face recognition using hog–ebgm. *Pattern Recognition Letters*, 29(10):1537–1543, 2008.

[112] O. Déniz, G. Bueno, J. Salido, and F. De la Torre. Face recognition using histograms of oriented gradients. *Pattern Recognition Letters*, 32(12):1598–1603, 2011.

[113] C. Sforza, G. Grandi, M. Binelli, D. G. Tommasi, R. Rosati, and V. F. Ferrario. Age-and sex-related changes in the normal human ear. *Forensic Science International*, 187(1):110.e1–110.e7, 2009.

[114] B. Moreno, A. Sanchez, and J. F. Velez. On the use of outer ear images for personal identification in security applications. In *IEEE International Carnahan Conference on Security Technology*, pages 469–476, 1999.

[115] Z. Mu, L. Yuan, Z. Xu, D. Xi, and S. Qi. Shape and structural feature based ear recognition. In *Advances in Biometric Person Authentication*, pages 663–670. 2004.

[116] M. Burge and W. Burger. Ear biometrics. In Pankanti S. Jain A.K., Bolle R., editor, *Biometrics*, pages 273–285. Springer, 1996.

[117] M. Burge and W. Burger. Ear biometrics for machine vision. In *Workshop of the Austrian Association for Pattern Recognition*, pages 275–282, 1997.

[118] D. J. Hurley, M. S. Nixon, and J. N. Carter. Force field feature extraction for ear biometrics. *Computer Vision and Image Understanding*, 98(3):491–512, 2005.

[119] C. Kyong, K. W. Bowyer, S. Sarkar, and B. Victor. Comparison and combination of ear and face images in appearance-based biometrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1160–1165, Sept 2003.

[120] H. J. Zhang, Z.C. Mu, W. Qu, L. M. Liu, and C. Y. Zhang. A novel approach for ear recognition based on ICA and RBF network. In *International Conference on Machine Learning and Cybernetics*, volume 7, pages 4511–4515, Aug 2005.

[121] A. Fabate, M. Nappi, D. Riccio, and S. Ricciardi. Ear recognition by means of a rotation invariant descriptor. In *International Conference on Pattern Recognition*, volume 4, pages 437–440, 2006.

[122] A. Sana, P. Gupta, and R. Purkait. Ear biometrics: a new approach. In *Advances in Pattern Recognition*, pages 46–50. 2007.

[123] W. Yu, Z. C. Mu, and H. Zeng. Block-based and multi-resolution methods for ear recognition using wavelet transform and uniform local binary patterns. In *International Conference on Pattern Recognition*, pages 1–4, Dec 2008.

[124] H. L. Zhao and Z. C. Mu. Combining wavelet transform and orthogonal centroid algorithm for ear recognition. In *IEEE International Conference on Computer Science and Information Technology*, pages 228–231, Aug 2009.

[125] P. Howland and H. Park. Generalizing discriminant analysis using the generalized singular value decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):995–1006, Aug 2004.

[126] A. Kumar and D. Zhang. Ear authentication using log-gabor wavelets. In *SPIE Defence and Security Symposium*, 2007.

[127] D. R. Kisku, H. Mehrotra, P. Gupta, and J. K. Sing. SIFT-based ear recognition by fusion of detected keypoints from color similarity slice regions. In *International Conference on Advances in Computational Tools for Engineering Applications*, pages 380–385, July 2009.

[128] C. Murukesh, A. Parivazhagan, and K. Thanushkodi. A novel ear recognition process using appearance shape model, Fisher linear discriminant analysis and contourlet transform. *Procedia Engineering*, 38:771–778, 2012.

[129] I. Omara, X. Wu, H. Zhang, Y. Du, and W. Zuo. Learning pairwise svm on deep features for ear recognition. In *International Conference on Computer and Information Science*, pages 341–346, May 2017.

[130] Z. Emersic, V. Struc, and P. Peer. Ear recognition: more than a survey. *Neurocomputing*, 255:26–39, 2017.

[131] L. Tian and Z. Mu. Ear recognition based on deep convolutional network. In *International Congress on Image and Signal Processing, BioMedical Engineering and Informatics*, pages 437–441, Oct 2016.

[132] Z. Mu. USTB ear image database. http://www1.ustb.edu.cn/resb/en/visit/visit.htm, 2009 (accessed October 1, 2017).

[133] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[134] Z. Emersic, D. Stepec, V. Struc, P. Peer, A. George, A. Ahmad, E. Omar, T. E. Boult, R. Safdari, Y. Zhou, S. Zafeiriou, D. Yaman, F. I. Eyiokur, and H. K. Ekenel. The unconstrained ear recognition challenge. In *International Joint Conference on Biometrics*, October 2017.

[135] A. Ahmad, E. Omar, and T. E. Boult. The unconstrained ear recognition challenge. In *International Joint Conference on Biometrics*, October 2017.

[136] C. C. Chang and C. J. Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27, 2011.

[137] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

Chapter A

DESCRIPTION OF THE IDENTIFICATION LFW DATASET

The original LFW dataset [3] consists of 5,749 subjects with images per subject ranging from 1 to 350 to yield a total of 13,233 images. This dataset, which was constructed for face matching rather than identification, is rearranged to generate a face identification dataset by only selecting the subjects with at least 5 images. Therefore, the resulting LFW identification dataset consists of 5,088 images corresponding to 255 subjects. The images are cropped to a size of 45 $\times$ 45 and frontalized using the method proposed in Hassner *et al.* [10].

In the identification LFW dataset, in addition to the original images, the LFW images are subjected to different types and levels of distortions, simulating visual quality distortions that can occur under real-world conditions, including impairments due to blur, block occlusion and noise. For blur and noise visual impairment types, the level of impairments were chosen such that the whole range of visual quality is represented from Poor (strong perceived impairment as compared to original source) to Excellent (no perceived impairment, original source). Four different levels of impairments are used for each impairment type, besides the original source. For the block occlusion impairment type, specific-location and random location contiguous block occlusions are added with three different sizes, in addition to the original source.

Addition of Distortions

Gaussian Blur

A Gaussian blur function is simulated as in (3.7) and convolved with the test face images of the three considered datasets. The filter size of the Gaussian blur kernel is set in number of pixels as [5, 5, 7, 9] for the different blur levels represented by the blur variance values [1, 2, 4, 8], respectively. The levels of distortions are carefully chosen to generate images covering a broad range of quality, from imperceptible levels to high levels of impairment. The dictionary is augmented with the same four Gaussian blur levels in addition to the original clean images.

Camera Shake Blur

Next unseen distortions, such as camera shake blur, are added. They are more general than the previously considered Gaussian blur, because the blur kernel is not symmetric. The 8 blur kernels provided by [1] are used, which were captured from a real camera, and are convolved with the test images of the three considered datasets. The blur kernels have different sizes including 13 $\times$ 13, 15 $\times$ 15, 17 $\times$ 17, 19 $\times$ 19, 21 $\times$ 21, 23 $\times$ 23 and 27 $\times$ 27. The before last 2 kernels have the same size but different form. These 8 blur kernels result from the relative motion of a camera mounted on a tripod (z-axis) with loosened x and y handles. The motion is an in-plane rotation (rotation around the z-axis), which is a significant component of human hand shake.

Block Occlusions

Several block occlusion sizes ranging from 10 percent to 50 percent are added, by replacing one or two blocks in each test image with one or two black boxes at major facial locations, including the eyes, the nose and the mouth. Therefore, the considered occlusions are either single or double. In addition, random location single

block occlusions are also added.

White Noise

Finally, the additive Gaussian noise is simulated with zero mean and variance σ^2. Four levels of white Gaussian noise are added to the face images, where the variance values are [5, 10, 20, 40]. Again, the levels of distortions are carefully chosen to generate images covering a broad range of quality, from imperceptible levels to high levels of impairment.