ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

FACE RECOGNITION WITH LOCAL WALSH TRANSFORM

Ph.D. THESIS

Meryem UZUN-PER

Department of Computer Engineering

Computer Engineering Program

FEBRUARY 2018

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

FACE RECOGNITION WITH LOCAL WALSH TRANSFORM

Ph.D. THESIS

Meryem UZUN-PER
(504112506)

Department of Computer Engineering

Computer Engineering Program

Thesis Advisor: Prof. Dr. Muhittin GÖKMEN

FEBRUARY 2018

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

YEREL WALSH DÖNÜŞÜMÜ İLE YÜZ TANIMA

DOKTORA TEZİ

Meryem UZUN-PER
(504112506)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Muhittin GÖKMEN

ŞUBAT 2018

Meryem UZUN-PER, a Ph.D. student of ITU Graduate School of Science Engineering and Technology 504112506 successfully defended the thesis entitled "FACE RECOGNITION WITH LOCAL WALSH TRANSFORM", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Muhittin GÖKMEN** Istanbul Technical University	.....
Jury Members :	**Prof. Dr. Zehra ÇATALTEPE** Istanbul Technical University	.....
	Prof. Dr. Fikret GÜRGEN Boğaziçi University	.....
	Assoc. Prof. Dr. Mustafa Ersel KAMAŞAK Istanbul Technical University	.....
	Asst. Prof. Dr. Berk GÖKBERK MEF University	.....

Date of Submission : **5 January 2018**
Date of Defense : **21 February 2018**

To my husband and my daughter,

FOREWORD

First of all, I would like to thank my advisor, Prof. Dr. Muhittin Gökmen, for his support and effort during my academic research. I believe that without his ideas, this thesis would never be finished. I would like to thank to members of my thesis committee, Prof. Dr. Zehra Çataltepe and Prof. Dr. Fikret Gürgen for their time and valuable comments. I thank to Assoc. Prof. Dr. Mustafa Kamaşak and Asst. Prof. Dr. Berk Gökberk for accepting to be a member of my defense committee and for their time.

I would like to thank Prof. Dr. Uluğ Bayazıt whose courses I have assisted for many semesters. He has always encouraged me to complete my academic researches. His support is priceless for me. I would also like to thank Assist. Prof. Dr. Yusuf Yaslan for encouraging me during my Ph.D. studies. I thank to Assoc. Prof. Dr. Hatice Köse and Prof. Dr. Nadia Erdoğan for their warm conversations and support.

My research in this thesis has been supported by several research funds and scholarships, including TÜBİTAK (the Scientific and Technological Research Council of Turkey) 2211-C PhD Scholarship for Priority Areas, TBD-TURKCELL Leaders of Technology Graduate Scholarship, and Istanbul Technical University Scientific Research Projects (İTÜ-BAP). I would like to thank the funding institutions for their support.

I would like to thank my dear roommates Mustafa, Doğan, and Gökhan for always being kind to me. I thank to Atakan for sharing his experience on the Ph.D. stuff warmheartedly. I would like to thank my dear assistant friends Dilara and Neziha for their support and love. I thank to Zeynep who is my lovely classmate in Ph.D. She has always encouraged me as a Ph.D. student mother. I cannot forget the supports of Emrahs. Emrah Başaran was the only person whom I could discuss the things about my thesis topic. Emrah Yiğit also supported me with his positive view of point. Lastly, my dear friend İmran always encouraged me to complete my Ph.D. studies.

Finally, I would like to thank my loving husband for his support and encouragement. I am sure that if my husband was not him, I could not complete this dissertation. Since 2010, he has always helped me by both motivating and doing housework. I would also like to thank my beautiful, dear daughter for being so nice, smart and well-behaved girl. I would also like to thank my mother, father, sisters, and brothers for being with me and supporting me. My mother-in-law, father-in-law and brother-in-laws have also been with me with their blessings.

February 2018 Meryem UZUN-PER

TABLE OF CONTENTS

Page

ABBREVIATIONS

AUC : Area Under Curve
BELBP : Block-wised Elliptical LBP
BWPCA : Block-based WPCA
CLWT : Cascaded Local Walsh Transform
CS-KDA : Class-specific Kernel Discriminant Analysis
DCT : Discrete Cosine Transform
DFT : Discrete Fourier Transform
FERET : The Face Recognition Technology
FLD : Fisher's Linear Discriminant Analysis
GGPP : Global Gabor Phase Patterns
HGPP : Histogram of Gabor Phase Patterns
HOG : Histograms of Oriented Gradient
HPEN : High-Fidelity Pose and Expression Normalization
ICA : Independent Component Analysis
L2ML : L2 Norm Metric Learning
LBP : Local Binary Patterns
LBPNet : LBP Network
LDA : Linear Discriminant Analysis
LFW : Labeled Faces in the Wild
LGBP : Local Gabor Binary Pattern Histogram Sequence
LGPP : Local Gabor Phase Patterns
LGXP : Local Gabor XOR Patterns
LPOG : Local Patterns of Gradients
LPQ : Local Phase Quantization
LWT : Local Walsh Transform
LZM : Local Zernike Moments
MBC : Monogenic Binary Coding
MBSIF : Multiresolution of Binarised Statistical Image Features
MLBP : Multi-scale LBP
MLPQ : Multiscale Local Phase Quantisation
MRF : Markov Random Field
PCA : Principal Component Analysis
PCLWT : Patch-based Cascaded Local Walsh Transform
PLWT : Patch-based Local Walsh Transform
PMH : Phase Magnitude Histogram
QBC : Quadrant Bit Coding
ROC : Receiver Operating Characteristic
SCface : The Surveillance Cameras Face Database
SDM : Supervised Descent Method
SIFT : Scale Invariant Feature Transform
SPAE : Progressive Auto-Encoders

SVD : Singular Value Decomposition
SVM : Support Vector Machine
WPCA : Whitened Principal Component Analysis
WT : Walsh Transform
XOR : Exclusive or
ZM : Zernike Moments

LIST OF TABLES

Page

LIST OF FIGURES

Page

FACE RECOGNITION WITH LOCAL WALSH TRANSFORM

SUMMARY

Face recognition is automatically identifying or verifying a person from a still image or a video frame. The challenges in this area largely occur due to illumination, viewpoint, expression, and resolution variances. There is a wide range of information security, surveillance, law enforcement, and entertainment applications which use face recognition despite there being more accurate biometric indicators such as fingerprint or iris. However, these indicators require the cooperation of those being observed whereas face images can be taken without any cooperation.

Face recognition systems generally have four main stages; face detection, alignment, feature extraction, and classification. In this dissertation, we focus on the feature extraction stage, which is probably the most challenging part of face recognition systems.

Local approaches have been used frequently due to their success in handling variations such as illumination and expression. Local representations are categorized into two types as sparse and dense. In sparse local representations such as Scale Invariant Feature Transform, points of interest are detected to be used for object detection. In dense local representations, such as Local Binary Patterns and Gabor, features are extracted by applying the method to each pixel of an image. In this dissertation, firstly, we propose a dense local representation which localizes the global Walsh Transform. Secondly, we propose methods to use this representation effectively.

The first contribution of this dissertation is a novel dense local representation called Local Walsh Transform (LWT). Walsh Transform (WT) kernel is defined as a set of orthogonal functions with values plus and minus ones. The dot product of any two distinct rows and columns of a WT kernel is zero. Transpose, conjugate, and inverse of a WT kernel are equal to itself. For the WT kernel of order N, if we denote rows by i from 0 to $N-1$, the i^{th} row has exactly i zero crossings.

In the LWT method, we use this special WT matrix to create filters and apply these filters to each pixels of an image separately. Thus, for a $N*N$ WT we create N^2 filters and so N^2 images, namely LWT components (ψ). Based on the fact that using phase information makes the recognition method somewhat more invariant to uniform illumination changes and blurring, we propose an approach to create complex images from LWT components. To this end, we utilize the symmetry between the LWT components with respect to the diagonal. By using this symmetrical relationship, we pair the symmetrical LWT components to create LWT complex images. LWT components which are on the upper side of the diagonal are considered as real part of the complex components, while the components which are on the lower side of the diagonal are considered as imaginary parts of complex images which are paired symmetrically with respect to the diagonal.

In addition to the LWT, we propose some additional methods to improve the performance of a face recognition system. The LWT is enhanced with some approaches such as Cascaded LWT (CLWT), dividing image into subregions, weighting, and reducing dimension of features by block-based dimension reduction method. In addition, LWT complex images are converted to grayscale images by XOR approach. For face verification problem, SVM is used for classification, while L2ML is used for dimension reduction.

Among the proposed LWT-based methods, Patch-based LWT (PLWT) method enlightens the abilities of the LWT most. In PLWT, we take the advantage of both sparse and dense local representations by applying the LWT method to salient points of face images. In feature extraction stage of the PLWT, first, landmarks of images are detected and some of them are selected. Then, the images and their landmarks may be scaled for once or twice. For each selected landmark at different scales of an image, a patch is created around the landmarks. Following this, the LWT method is applied to all patches of the image and LWT complex images are created. Each complex image is divided into non-overlapping subregions and Phase Magnitude Histogram (PMH) of each subregion is calculated. The PMHs of the subregions are concatenated to create the features of complex images and these features are concatenated to create the features of the patches. Dimensions of the features of the patches are reduced by Whitened Principal Component Analysis method. We choose an unsupervised dimension reduction method to make our methods completely unsupervised and be operable on the databases which do not contain class labels of images. For a test image, the same algorithm is applied to the image and dimensions of the patches are reduced by projection matrices learned in the training stage. Identity of the image is determined according to the similarity between the features of the test image and that of the training set images.

The proposed PLWT method is further improved by the CLWT approach. In the CLWT, we apply LWT again to both real and imaginary parts of the LWT complex images. In other words, we once more apply the LWT to sharpened images in which shape information is extracted by the first application of the LWT. This approach gives a depth to the method and increases success rate especially on low resolution images in which shapes and edges are blurred. To use this approach in the PLWT method, all needed is to apply the CLWT instead of the LWT. When the proposed method is carried out with the CLWT approach, we call this method Patch-based CLWT (PCLWT).

We show the performance of the proposed methods, by applying them to challenging, widely known, and popular benchmark databases. Specifically, for the face identification problem the Face Recognition Technology (FERET) and the Surveillance Cameras Face (SCface) databases are used, for the face verification problem the challenging Labeled Faces in the Wild (LFW) database is used. The FERET is one of the most widely used benchmarks in the evaluation of face recognition methods. This database gives an opportunity to evaluate methods with various facial expressions, lighting conditions, cameras, scale, and aging property. The images of SCface are taken in an uncontrolled indoor environment that mimics the real world conditions of the indoor face recognition problem. It contains very low resolution images which are taken in the dark. The LFW is a database designed to facilitate the benchmarking of unconstrained pair matching and is also intended to illustrate the problem of learning from one example. This database consists of images collected

from the web which contain large pose, occlusion, and expression variations. For most of our experiments, we use the "unsupervised" protocol of the LFW since during both training and testing phases we do not use label information of the subjects, and not access to whether any pair of images is of the same subject. For methods which include SVM or L2ML methods, we use "image restricted with no outside data" protocol of the LFW.

In the preprocessing step, images of databases are normalized and aligned by the frontalization method, a similarity transform or the High-Fidelity Pose and Expression Normalization (HPEN) with a similarity transform. The landmarks of the face images are detected by using the Supervised Descent Method (SDM) or Dlib. To the best of our knowledge, the PLWT and PCLWT method achieve the best performance on both the FERET and the SCface databases, and the second best unsupervised category result on the LFW database. Although the other proposed methods are also successful, they are below the state-of-the-art results.

In conclusion, the LWT is a strong image descriptor which reveals the shape characteristics of images. The CLWT is especially effective on low resolution images where shape characteristics are blurred. Experimental results show that both the PLWT and PCLWT are robust in terms of illumination, expression, occlusion, and low resolution. The high performances of the proposed methods show that these methods are effective on both wild and low resolution images, and are good alternatives to unsupervised hand-crafted methods.

YEREL WALSH DÖNÜŞÜMÜ İLE YÜZ TANIMA

ÖZET

Yüz tanıma problemi, durgun görüntü veya videolardan elde edilmiş görüntülerdeki kişilerin otomatik olarak kimliklerinin tespit edilmesi veya doğrulanmasıdır. Bu alandaki sorunlar, görüntülerin aydınlatma, çözünürlük, yüz ifadesi veya poz gibi farklılıklarından kaynaklanmaktadır. Yüz tanıma sistemleri bilgi güvenliği, otomatik denetleme, eğlence uygulamaları gibi birçok alanda kullanılmaktadır. Yüzden daha kesin sonuçlar üretebilecek biyometrik göstergeler (ör: parmak izi, iris) vardır. Ancak, bu göstergeler alınırken verici ile işbirliği yapmayı gerektirmektedir. Yüz imgelerinin verici farkında olmasa dahi alınabilmesi, yüz tanıma sistemlerine olan ilgiyi arttırmıştır.

Yüz tanıma sistemleri genel olarak dört aşamadan oluşur. Bunlar, yüzün tespit edilmesi, normalizasyon, öznitelik çıkarımı ve sınıflandırmadır. Yüz tespiti, normalizasyon ve sınıflandırma algoritmaları ne kadar başarılı olursa olsun, öznitelik çıkarımı aşaması başarılı değilse o sistem istenilen başarıya ulaşamayacaktır. Bu tezde başarılı bir yüz tanıma sistemi geliştirebilmek için belki de en zorlu aşama olan öznitelik çıkarımı aşaması üzerinde durulmuştur.

Poz ve aydınlatma farklılıklardan daha az etkilenen yerel yaklaşımlar üzerinde sıkça çalışılmaktadır. Yerel betimleyiciler, seyrek ve yoğun yöntemler olarak ikiye ayrılmaktadır. Seyrek yerel betimleyiciler arasında yer alan Ölçek Değişimsiz Öznitelik Dönüşümü (ing. Scale Invariant Feature Transform) yönteminde öncelikle ana noktalar belirlenip, bu noktalar etrafında alınan yamalardan öznitelik çıkarılır. Yerel İkili Örüntüler (ing. Local Binary Patterns) ve Gabor gibi yoğun yerel betimleyicilerde ise yöntem tüm piksellere ayrı ayrı uygulanarak öznitelik çıkarılır. Bu iki yerel yaklaşımı bir arada kullanan hibrit yöntemler de vardır. Bu tezde, öncelikle global Walsh Dönüşümünü (WD) temel alan bir yoğun yerel betimleyici önerilmiştir. Ardından, bu betimleyicinin etkili bir şekilde kullanılmasını sağlayan yöntemler önerilmiştir.

Bu tezin literatüre olan ilk katkısı yeni yoğun yerel betimleyici olan Yerel Walsh Dönüşümüdür (YWD). Walsh Dönüşümü (WD) matrisi çarpanları ± 1 olan ortogonal fonksiyonlardan oluşur. Gerçel ve simetriktir yani matris, transpozesi ve eşleneğine eşittir. WD matrisinde herhangi bir satırın, kendisi dışında herhangi bir satırın transpozesiyle veya sütunla skaler çarpımı sıfırdır. $N*N$ boyutlu WD matrisinde, i 0'dan N-1'e satırları numaralandırsın; i.nci satır her zaman i adet sıfırdan geçiş içerir.

YWD yönteminde, WD matrisi imgenin her pikseline ayrı ayrı uygulanarak $N*N$ boyutlu WD için N^2 adet imge, yani YWD bileşeni (ψ) üretilir. Faz bilgisini kullanmak yöntemi aydınlatma ve bulanıklık değişimlerine karşı dayanıklı yapar. Faz bilgisini kullanabilmek için karmaşık imgelere ihtiyaç vardır ancak YWD yalnızca gerçel imgeler üretir. YWD bileşenlerinden karmaşık imgeler üretebilmek için bu bileşenler

incelenerek bir yöntem önerilmiştir. Bu yöntemde, YWD bileşenlerinin birbirlerine olan simetrisi kullanılarak, N*N boyutlu WD matrisi için $(N*N-N)/2$ adet karmaşık imge oluşturulur. Uygulanan WD'nin satır ve sütunları sırasıyla u ve v ile gösterilirse, $u < v$ olmak şartı ile, (u,v) filtresi ile üretilen YWD bileşeni karmaşık imgenin reel kısmını, (v,u) filtresi ile üretilen YWD bileşeni karmaşık imgenin sanal kısmını oluşturur. YWD karmaşık imagelerini üreten bu yöntem aşağıdaki gibi tanımlanabilir:

$$Z_{u,v} = kompleks(\psi_{u,v}\,,\,\psi_{v,u}) = \psi_{u,v} + i\,\psi_{v,u} \;\;;\;\; \forall(u,v = 0,...,N-1;\; u < v). \qquad (1)$$

Şekil 3.1'de, YWD yönteminin 4x4 WD kullanılarak bir imgeye uygulanması sonucu üretilmiş YWD bileşenleri gösterilmektedir.

YWD yoğun yerel betimleyicisi önerildikten sonra bu betimleyiciyi etkili bir şekilde kullanacak çeşitli yöntemler önerilmiştir. Bunların bir kısmı, iki aşamalı YWD (Kaskatlanmış YWD, KYWD) yöntemi, histogram hesaplama öncesinde imgeyi alt bölgelere ayırma, ağırlıklandırma ve imgeyi bloklara ayırıp onlardan elde edilen özniteliklerin boyutlarını azaltma yöntemleridir. Bunlar dışında, YWD betimleyicisinin imgelerdeki nirengi noktalarının etrafında uygulanması ve YWD karmaşık imgelere XOR yöntemi uygulayıp gri seviye imgelere dönüştürülmesi ile de geliştirilmiştir. Ayrıca, yüz doğrulama problemi için Destek Vektör Makinaları (DVM, ing. Support Vector Machines) yöntemi sınıflandırma aşamasında, L2 Norm Metrik Öğrenme (L2MÖ, ing. L2 Norm Metric Learning) yöntemi boyut indirgeme aşamasında kullanılmıştır.

Bu yöntemler arasında en etkililerinden biri, YWD betimleyicisinin imgelerdeki nirengi noktalarının etrafına uygulanması ile elde edilmiştir. Bu yöntemde, önce yüze ait nirengi noktaları tespit edilir. Tespit edilen bu noktaların bir kısmı seçilerek etraflarında yamalar kesilir. İstenilirse imgenin çeşitli boyutları kullanılıp tespit edilen nirengi noktaları etrafında aynı boyutlu yamalar kesilir. Burada imgenin farklı boyutları etrafında aynı boyutlu yama kesmek o bölge için hem yerel hem de globale yakın özelliklerinin tespiti için etkilidir. Bu yamalara YWD yöntemi uygulanır ve elde edilen karmaşık imgelerin histogramı hesaplanır. Bu yöntem ile eğitim kümesindeki tüm imgelerin yamalarının öznitelikleri çıkarılır. Bu şekilde çıkarılan yamaların öznitelikleri Beyazlatılmış Temel Bileşenler Analizi (ing. Whitened Principal Component Analysis) yöntemi ile indirgenir. Bu çalışmada, önerilen yöntemin gözetimsiz yöntem olması ve imgelerin sınıf etiketlerini içermeyen veritabanları üzerinde de çalışabilmesi için gözetimsiz boyut indirgeme yöntemi tercih edilmiştir. Test aşamasında, test imgesinin özniteliği bu yöntemle çıkarıldıktan sonra en benzer özniteliğe sahip eğitim kümesi imgesinin kimliği test imgesinin kimliği olarak belirlenir.

Ayrıca, önerilen yöntem KYWD yaklaşımı ile geliştirilmiştir. Bu yaklaşımda, YWD yöntemi imgeye uygulandıktan sonra üretilmiş olan karmaşık imgelerin sanal ve gerçel kısımlarına YWD yöntemi tekrar uygulanır. Bu yaklaşım yönteme bir derinlik katar ve özellikle düşük çözünürlüklü imgelerde başarımı artırır. YWD'nin nirengi noktaları etrafında uygulanması yönteminde YWD yerine KYWD yöntemi de uygulanabilmektedir.

Önerilen yöntemler uluslararası bilinen ve çeşitli zorluklar içeren veritabanları üzerinde test edilmiştir. Bu veritabanlarından FERET ve SCface yüz tanıma problemi için, LFW yüz doğrulama problemi için kullanılmıştır. Bunlardan FERET yarı kontrollu ortamda çekilmiş, yüz ifadesi, aydınlatma, kamera, boyut gibi farklılıklar

barındıran imgeler içermektedir. SCface gerçek dünyayı yansıtan, karanlık ortamda çekilmiş imgeler dahil kontrolsuz iç mekan ortamında çekilmiş düşük çözünürlüklü imgeler içermektedir. LFW veritabanı ise internet sitelerinden toplanmış ifade, poz, imgelerin bir kısmının kapanması, aydınlatma, çözünürlük gibi çeşitli zorluklara sahip imgeler içermektedir. Önerdiğimiz çoğu yöntem gözetimsiz yöntem olduğu için LFW'nun "gözetimsiz" protokolünde, yani imgelere ait hiçbir bilginin bilinmediği sınırlı sayıda imgenin bulunduğu kategoride sonuçlar verilmiştir. DVM ve L2MÖ metodlarını içeren yöntemlerimizin sonuçları LFW'nun yalnızca LFW tarafından önerilen etiketli çiftlerinin bulunduğu dışarıdan imgenin kullanılmadığı kategoride verilmiştir.

Önişleme aşamasında, imgeler frontalizasyon, benzerlik dönüşümü ya da HPEN (High-Fidelity Pose and Expression Normalization) yöntemleri ile hizalanmış ve normalize edilmiştir. Yüzlere ait nirengi noktaları SDM (Supervised Descent Method) ya da Dlib ile tespit edilmiştir. Nirengi noktaları etrafında öznitelik çıkarımı ve ardından boyut idirgeme yöntemi için yapılan deneyler sonucunda FERET ve SCface için bilinen en yüksek başarım elde edilirken, LFW veritabanının "gözetimsiz" protokolüne göre ikinci en yüksek başarım elde edilmiştir. Önerilen diğer yöntemlerin başarımı iyi olsa bile bilinen en yüksek başarımların altındadır.

Yapılan deneyler, YWD yönteminin şekil özelliklerini çıkaran güçlü bir imge betimleyici olduğunu, KYWD'nin de özellikle düşük çözünürlüklü imgelerde oldukça etkili olduğunu göstermiştir. YWD ve KYWD'nin nirengi noktalarında uygulanması yöntemi ile elde edilen yüksek başarımlar yöntemin aydınlatma, ifade, yüzün bir kısmının kapanması ve düşük çözünürlük sorunlarına karşı dayanıklı olduğunu göstermektedir. Ayrıca, bu yöntem yalnızca yüz tanıma problemine değil tüm biyometrik tanıma problemlerine uygulanabilir.

1. INTRODUCTION

Face recognition is automatically identifying or verifying a person from a still image or a video frame. It has been studied for more than 30 years but is still a challenging subject of computer vision. The challenges in this area largely occur due to illumination, viewpoint, facial expression, scale, and resolution variances. An image just contains intensity values of pixels. It is seen that variations of two different faces in the same condition is smaller, most of the times, than the variations of two face images of a same person in different conditions such as illumination, expression, and viewing point [1].

As technology evolves, the requirement of face recognition application increases. There is a wide range of information security, surveillance, law enforcement, and entertainment applications which use face recognition despite there being more accurate biometric indicators such as fingerprint or iris [1]. However, these indicators require the cooperation of those being observed whereas face images can be taken without the requirement for any such cooperation. This property has also increased interest in face recognition as a field of study. 2D face image recognition is mostly preferred to the 3D face recognition since acquisition of 2D images are easier, cheaper, and faster than that of 3D images [2]. Its potential effect on surveillance issue such as searching for wanted criminals, suspected terrorists, and missing children cannot be underrated. Some of the face recognition applications can be listed as follows [1]:

- Law enforcement: CCTV control and surveillance, post-event analysis, shoplifting, suspect tracking and investigation.
- Information security: TV parental control, TV person identification for deaf person, personal device logon, file encryption, intranet security, medical records, secure trading terminals.
- Smart cards: Driving licenses, entitlement programs, immigration, national ID cards, passports, voter registration.

- Entertainment: Social media applications, video games, virtual reality, training programs, human robot interaction, human computer interaction.

Face recognition systems generally have four main stages; face detection, preprocessing, feature extraction, and classification. Face detection is the first step of face recognition to find and locate faces of humans in images automatically. After this step, preprocessing operations may be applied. Preprocessing step may include normalization, histogram equalization, masking, filtering, correction, smoothing, sharpening, following these by face alignment using just eyes or all detected landmarks of the face images, and cropping. The last step of the face recognition is the classification step in which the identities of the face images are determined. The most frequently used methods to determine the identity of the people are k-nearest neighbor method and Support Vector Machine (SVM) together with several distance metrics such as Euclidean distance, cosine similarity, chi square test, and histogram intersection.

The process of feature extraction can be categorized into two classes as global and local. In global approaches, features are extracted from the entire image. Global approaches include subspace approaches such as Principal Component Analysis (PCA) [3], and Linear Discriminant Analysis (LDA) [4]; the transformation from spatial space to frequency space as in Discrete Cosine Transform (DCT) [5], and Discrete Fourier Transform (DFT) [6]; as well as moment based approaches such as Zernike Moments [7]. Local approaches have been used more frequently due to their success in handling variations such as illumination and expression. Local representations are also categorized into two types as sparse and dense representations [8]. In sparse local representations such as Scale Invariant Feature Transform (SIFT) [9], points of interest are detected to be used for object detection. In dense local representations, such as Local Binary Patterns (LBP) [10] and Gabor [11], features are extracted by applying the method to each pixel of an image.

In this thesis, we focus on the feature extraction stage, which is probably the most challenging part of face recognition systems. Even though the most successful face detection, alignment, and classification algorithms are used, if the feature extraction algorithm does not perform adequately, the system will not be successful. In the thesis,

we propose feature extraction methods to solve illumination, expression, occlusion, time-lapse variations, and low resolution problems.

In face recognition problem, one of the challenges is problem of learning from one example [12]. In this problem, just one image of people are provided in the gallery set to determine the identity of a test image. In most of the real world cases we need to learn from just one example. Beside this, in some cases gallery set may not include an image of the test person. In order to struggle with these problems, we propose an unsupervised feature extraction method which will be successful in face identification problem if there is only one example of people, and in face verification problem even if there is no example of the people in the training stage.

1.1 Contributions of the Dissertation

The major contributions of this dissertation can be summarized as follows:

- The first contribution of this dissertation is Local Walsh Transform (LWT) method which is a novel unsupervised dense local image representation method. The LWT method is based on the well-known Walsh Transform (WT). It localizes the global WT by applying it to each pixel of an image and decomposing the image into multiple components, and then producing LWT complex images by using the symmetrical relationship between them. This method is a simple feature extraction method like Local Binary Patterns (LBP) [10] and Gabor [11]. In this dissertation, we show the effect of this method on the face recognition problem.

- The second contribution of this dissertation is Cascaded LWT (CLWT) method, which is also a dense local image representation method. It applies LWT again to real and imaginary parts of LWT complex images. Applying the LWT once more to real and imaginary parts of LWT complex images increases the success rate especially on low resolution images. We show the effect of this method on the face recognition problem.

- Other contributions of this dissertation are Patch-based LWT (PLWT) and Patch-based CLWT (PCLWT) methods which use efficiently the proposed LWT and CLWT methods, respectively. These methods take advantages of both sparse and dense local image representations by applying the LWT or the CLWT to

patches extracted around landmarks of multi-scaled face images. The extracted high dimensional features of the patches are reduced through the application of the Whitened Principal Component Analysis (WPCA).

- Finally, we show the effect of the proposed PLWT and PCLWT methods on the challenging, widely known databases which are the Face Recognition Technology (FERET) [13], Labeled Faces in the Wild (LFW) [12], and the Surveillance Cameras Face (SCface) [14] databases. The FERET database contains images including different facial expressions, lighting conditions, cameras, scale, and aging variances in a semi controlled environment. The LFW database consists of wild images collected from the web which contain variability in terms of pose, lighting, focus, resolution, facial expression, age, gender, race, accessories, make-up, occlusions, background, and photographic quality. The SCface database mimics the real world conditions of the indoor face recognition problem. As a result of experiments the state-of-the-art performance is achieved on the FERET and SCface databases, and the second best unsupervised category result is achieved on the LFW database.

Other than these contributions, we propose some other methods which lead to lower performance. One of these methods is LWT with block-based dimension reduction. In this method, after extracting features by applying the LWT to all pixels of an image, blocks of the features are reduced by WPCA (Block-based WPCA (BWPCA)) method. In this method, feature extraction process is enriched by dividing images into subregions and weighting them.

Another proposed method is the LWT with XOR approach. In this method, after creating LWT complex images of an image by the LWT or the CLWT, phase values of each pixels are calculated and quantized. Then, XOR operator is applied to these quantized images and grayscale images of complex images are created. Histogram of these grayscale images give the features of the image. This method is also improved by weighting, dividing into subregions and BWPCA approaches.

In the classification step of face verification problem, as an alternative SVM is used in order to decide if images are matched or unmatched. In this case, only classification step in the PLWT or in the PCLWT methods is changed by SVM. In addition, in the

PLWT and the PCLWT methods, instead of WPCA method, L2 Norm Metric Learning (L2ML) dimension reduction method is applied.

1.2 Thesis Outline

The rest of the dissertation is organized as follows:

In Chapter 2, we review the studies on local representations, WT-based approaches, and some preprocessing methods.

In Chapter 3, the LWT method is explained in detail. The databases, namely the FERET, the LFW and the SCface, which are used in the experiments are introduced. Experimental results for the LWT method is given and complexity of the proposed method is discussed.

In Chapter 4, the proposed LWT is improved with some approaches such as weighting, separating into subregions, and Cascaded LWT (CLWT). The LWT with block-based dimension reduction, LWT with XOR approach, Patch-based LWT (PLWT) and Patch-based CLWT (PCLWT) methods are introduced. Experimental results are given for each method. Alternatively, for face verification problem, SVM is used as a classification method, and L2ML method is used as dimension reduction method. Complexity of the PLWT and the PCLWT methods are discussed.

In Chapter 5, the thesis is concluded and possible future directions are given.

2. LITERATURE SURVEY

In this chapter, first, we give literature review of some local representations, then give properties of WT matrix and summarize works on WT, and finally explain some frontalization and normalization methods.

2.1 Local Representations

Local representations of images arouse interest due to their success in handling variations. Among dense local representations, Local Binary Patterns (LBP) is the most frequently used local representation [10, 15]. LBP, which was originally designed for texture description by Ojala et al. [15], has been used for face recognition by Ahonen et al. [10]. In this method, a binary number is determined for every pixel of an image by thresholding the pixel with its eight-neighbors. If neighbor pixel is smaller than center pixel it assigns 0 otherwise 1 and creates a binary number for each pixel of the image. These binary numbers are then used to create description of an image. Figure 2.1 shows an illustration of the LBP operator [10]. In [10], Ahonen et al. also introduce a weight map for a face. For example for face recognition, eyes are more distinctive than cheeks. Figure 2.2 shows a face divided into 7*7 subregions and shows weight of each subregion in which weight of a black cell is 0, dark grey is 1, light grey is 2, and white is 4. In [10], the description image is divided into 7*7 non-overlapping subregions, and after calculating the histogram of each subregion separately, weighted histograms are concatenated to create the final feature vector of the image. Many works use the LBP and its variants such as Local Ternary Patterns [16], Multi-scale LBP (MLBP) [17], and LBP Network (LBPNet) [18].

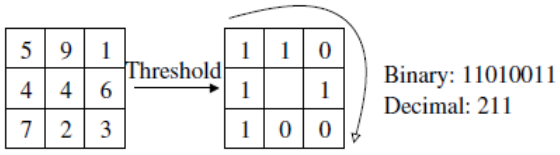


Figure 2.1 : Illustration of the LBP operator [10].

(a) (b)

Figure 2.2 : (a) Dividing image into 7*7 subregions. (b) The weight map proposed in [10].

2D Gabor filters [11] which produce Gabor phases and magnitudes at several orientations and scales are frequently used in face recognition methods. Some of the works which use Gabor filters are Local Gabor Binary Pattern Histogram Sequence (LGBP) [19], Histogram of Gabor Phase Patterns (HGPP) [20], and Local Gabor XOR Patterns (LGXP) [21]. In LGBP, after applying Gabor filters to an image, LBP is applied to Gabor features and so LGBP maps are created. Histograms of the LGBP maps are calculated by dividing them into non-overlapping subregions and then these histograms are concatenated to create final histogram sequence of the image. Figure 2.3 summarizes the approach used in [19].

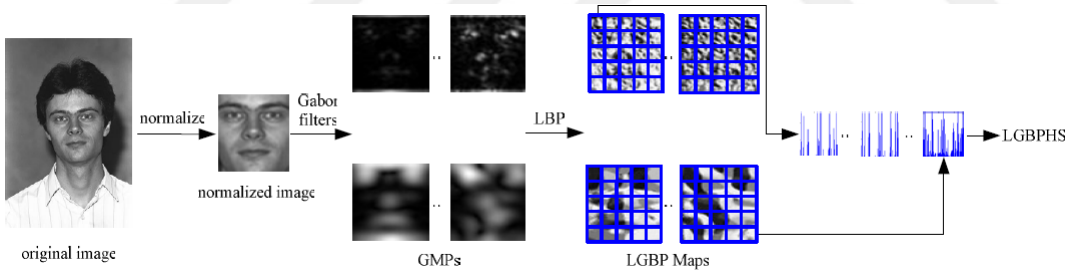


Figure 2.3 : Illustration of the LGBP method [19].

A more advanced method which uses Gabor is HGPP [20]. In this method, after applying Gabor filters to an image, Dougman's Quadrant Bit Coding (QBC) method is applied to the resulting complex Gabor images. According to QBC, a pixel is labeled as zero if the pixel value is greater than zero, and otherwise labeled as one. So, each pixel of real and imaginary parts of Gabor images are constituted by 0 or 1. Then, [0-256] valued images are created by using eight neighbors of each pixel of these images. This approach is called as Global Gabor Phase Patterns (GGPP). Another approach used in the paper is Local Gabor Phase Patterns (LGPP). In this approach, after applying QBC method, "exclusive or (XOR)" operator is applied to each pixel

with its eight neighbors and [0-256] valued images are created. Figure 2.4 shows an illustration of this operation [20]. Then these GGPP and LGPP features are divided into non-overlapping subregions and histogram of these regions are calculated. These histograms constitute the features of the image which is called as HGPP. Figure 2.5 summarizes feature extraction procedure of the HGPP method.

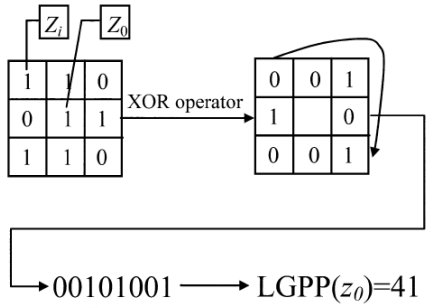


Figure 2.4 : Illustration of the LGPP method [20].

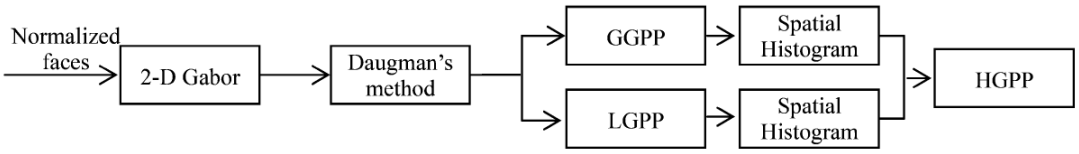


Figure 2.5 : Overview of the HGPP method [20].

Another approach which uses Gabor wavelets is proposed by Xie et al [21]. In this approach, after applying Gabor filter to an image, the Gabor complex images are quantized into four phase angle ranges (ie. $(0^\circ - 90^\circ], (90^\circ - 180^\circ], (180^\circ - 270^\circ], (270^\circ - 360^\circ]$. If pixel value of the complex image is between 0° and 90° "0" is assigned instead of this value. If the value is between 90° and 180° "1" is assigned instead of this value and so on. After this quantization operation, XOR method is applied as in [20]. This method is called as Local Gabor Xor Patterns (LGXP). After obtaining LGXP maps by this method, feature extraction step starts.

In the feature extraction step, direct use of histogram is not effective due to high dimension of features and "small sample size" problem. So, Block-based Fisher's linear discriminant (BFLD) approach is proposed to solve this problem [21]. In this method, LGXP maps of an image are divided into non-overlapping blocks and then each block is divided into non-overlapping subregions. Initially, histograms of subregions of a block is calculated. Then these histograms are concatenated to

constitute histogram of a block. For each LGXP maps of an image, histogram of each block is calculated. Then, histograms of each corresponding blocks are concatenated. For example, top left block of all LGXP maps of an image are concatenated to create the histogram of top left block of the image. Figure 2.6 depicts this feature extraction procedure of the algorithm for an image. Same operations are applied to all blocks of LGXP maps to constitute histogram of each block of the image. This feature extraction procedure is applied to all images in the training set. Then features are represented as "image vs. block" matrices. FLD dimension reduction method is applied to these matrices and projection matrices are learned. Then the blocks are projected to new dimension by these projection matrices. The projection matrices are stored to be used in testing stage. In the testing stage, for each test image after calculating histograms of blocks, they are projected to new dimensions by the related projection matrices. Finally, the identity of the test image is determined by using similarity between features of the test image and that of the training images.

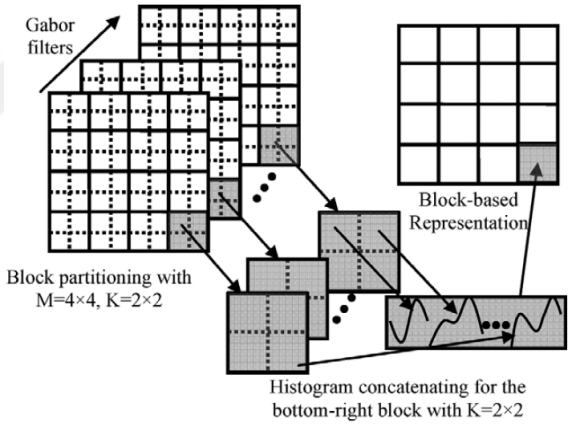


Figure 2.6 : Application of BFLD method after obtaining LGXP maps of an image [21].

Local Zernike Moments (LZM) method is based on Zernike Moments (ZM) which is successful on fingerprint recognition [22]. LZM method is the application of ZM to each pixel of an image. After obtaining LZM components from the first application of the LZM, it is applied again to the LZM components. It is asserted that second application of the LZM reveals statistics of shape characteristics which is obtained from the first application of it. Then, all LZM components are divided into subregions which are overlapping by half of the subregions and histograms of each subregions

are calculated separately. Finally, the histograms are concatenated to construct final feature vector of an image.

In Monogenic Binary Coding (MBC) [23], after applying multiscale log-Gabor filters as a band-pass filter, monogenic signal representation is applied which is built around the Riesz transform which is a natural multidimensional extension of the Hilbert transform. Monogenic signal representation decomposes an image into amplitude, phase, and orientation components, which represent the signal energetic, structural, and geometric information, respectively. After obtaining amplitude, phase, and orientation of an image, LBP-like operations are applied to each component separately. Histogram of each component is calculated after dividing them into non-overlapping subregions. For face recognition these features are compared with cosine similarity. As a second method in this paper, Xie's BFLD method is applied to MBC features of the image. After that, a weighting approach is used for fusion of the features. In the experiments it is shown that MBC methods run faster than Gabor-based approaches and produce competitive results with them.

In [24], after applying LBP-like operator to take difference of pixels, Fisher criterion is applied to maximize the ratio of between-class scatter and within-class scatter. Soft Sampling Matrix, image filter projections and dominant patterns are learned by applying this criterion. For each pixel, extracted pixel difference matrix is projected with the learned image filters and soft sampling matrix. The resultant submatrix is then transformed to form a Discriminant Pattern Vector (DPV). Finally, the dominant pattern which is the most similar with the extracted DPV determines the identity of the image. The method is also improved by WPCA.

In Scale Invariant Feature Transform (SIFT) method, which is a sparse local representation method, the scale space of an object image is constructed and then difference of Gaussian is applied to extract keypoints [9]. After determining the keypoints, their orientations are assigned and these orientations are compared to locate an object within the image. There are also some studies which use the advantages of both sparse and dense local representations. In [25], SIFT is applied to patches extracted from five scales of images by visiting every pixel of these images. In [26], overlapping multiple scales of face image patches centered around facial landmarks are extracted. Features of each patch are extracted by LBP, SIFT, Histograms of

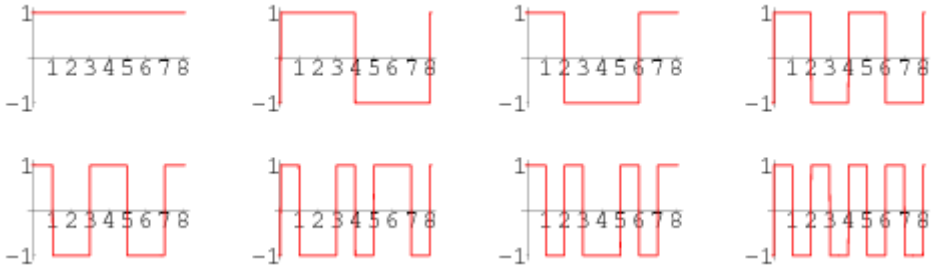
Oriented Gradient (HOG), and Gabor. In [27], features of overlapping multiple scales of face image patches are determined by LBP or SIFT. Following this, a Gaussian mixture model is built on the features before face verification. These works show that using high dimensional data by taking the advantages of both sparse and dense representations increases the success rate.

2.2 Walsh Transform

Walsh Transform (WT) kernel is defined as a set of orthogonal functions with values plus and minus ones [28]. The WT consists of rectangular or square waveforms with values +1 or -1. Figure 2.7 (a) and 2.7 (b) show waveforms of WT of order 4 and 8 respectively [29]. Properties of the WT matrix:

- WT matrix is constituted of just ± 1s.
- The first row and the first column of a WT matrix are always +1.
- It is real and symmetric i.e. the matrix is equal to its transpose and conjugate.
- The dot product of any two distinct row and column is zero, i.e. they are orthogonal.
- It can be constructed for number of rows and columns which are power of 2.
- For WT of order N, let j denotes rows from 0 to $N-1$, j^{th} row has exactly j zero crossings.

(a)



(b)

Figure 2.7 : (a) Waveforms of WT of order 4. (b) Waveforms of WT of order 8.

Hadamard matrix supplies all the properties of WT matrix after rearranging the rows of the matrix [30]. The lowest Hadamard matrix is order of two as follows:

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{2.1}$$

Hadamard matrix of order $N = 2^n$ can be constructed by Kronecker product of 2*2 Hadamard matrix and $\frac{N}{2} * \frac{N}{2}$ Hadamard matrix. The application of Kronecker product and so construction of a Hadamard matrix of order N is as follows:

$$H_{2^n} = H_2 \bigotimes H_{2^{n-1}} = \begin{bmatrix} H_{2^{n-1}} & H_{2^{n-1}} \\ H_{2^{n-1}} & -H_{2^{n-1}} \end{bmatrix}. \tag{2.2}$$

Figure 2.8 shows Hadamard matrix of order 8. In order to convert a Hadamard matrix to a WT matrix, number of sign changes (zero crossings) in rows of the Hadamard matrix are counted. The number of the sign change in a row of the Hadamard matrix gives the order of the row in WT matrix. In Figure 2.8, the last column represents the number of the sign changes in the related row. In Figure 2.9, the matrix is rearranged in the increasing order according to the number of the sign changes. This matrix is also called as Walsh-Hadamard Transform (WHT).

$$H_{2^3} = \left|\begin{array}{cccccccc|c} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 7 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 3 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 4 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 6 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 2 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 5 \end{array}\right|$$

Figure 2.8 : Hadamard Matrix of order 8.

$$H_{2^3 ordered} = \left|\begin{array}{cccccccc|c} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 2 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 3 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 4 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 5 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 6 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 7 \end{array}\right|$$

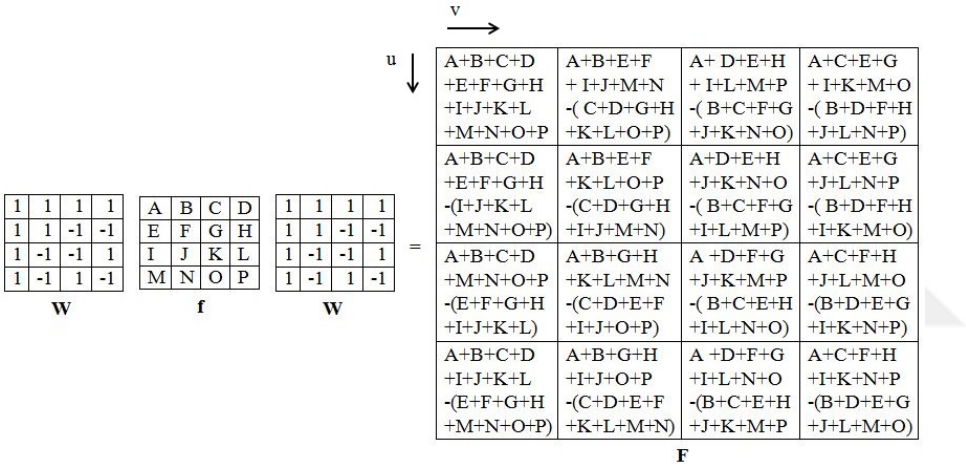
Figure 2.9 : Walsh-Hadamard Transform of order 8.

WT can be applied to an image as follows:

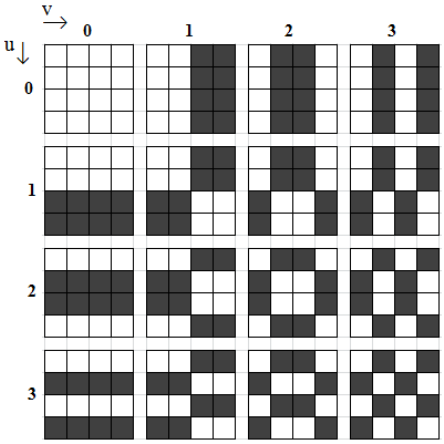
$$\mathbf{F} = \mathbf{W}\,\mathbf{f}\,\mathbf{W} \tag{2.3}$$

where **f** is the image, **W** is the WT kernel and **F** is the transformed image. **f**, **W**, and **F** are square matrices and their sizes must be equal. In order to fit the dimension of the image to the kernel size, the image have to be resized by either zero padding or interpolation. This resizing operation deforms the image, and reduces the success of methods [31].

Figure 2.10 (a) shows the application of a 4*4 WT to a 4*4 image symbolized by letters, and the output matrix shows how each pixel is constructed. Figure 2.10 (b) visualizes filters for a 4*4 WT kernel which constructs the transformed image in which the white squares symbolize plus and dark squares symbolize minus. Figure 2.11 (a) shows an image which is resized to 256*256 and Figure 2.11 (b) shows the transformed image which is obtained by applying 256*256 WT to the image.



(a)



(b)

Figure 2.10 : (a) Application of WT to a matrix symbolized by letters. (b) 16 direct filters to produce application result (**F**) for a 4*4 WT kernel (**W**).

(a) (b)

Figure 2.11 : (a) An image with size 256*256. (b) The transformed image with 256*256 WT kernel.

The original image can be reconstructed lossless from the transformed image by using the same size WT kernel in the following formula:

$$\mathbf{f} = \frac{1}{N^2}\,\mathbf{W}\,\mathbf{F}\,\mathbf{W} \tag{2.4}$$

where **F** is the transformed image, **W** is the WT kernel, **f** is the reconstructed image which is same as the original image, and N is the order of the WT kernel.

Alternatively, WT can be applied to an image directly without creating a WT kernel [30]. By this method, each pixel of the transformed image is calculated separately and so, unnecessary pixels of the transformed image may not be created. The application of WT of order $N = 2^n$ to an image is as follows:

$$F(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)(-1)^{q(x,y,u,v)} \tag{2.5}$$

where $u \in \{0,...,N-1\}$, $v \in \{0,...,N-1\}$, $f(x,y)$ represents the intensity values of an image over a N*N matrix, and

$$q(x,y,u,v) = \sum_{i=0}^{n-1} (p_i(u)x_i + p_i(v)y_i) \tag{2.6}$$

where the terms x_i and y_i are the digits of the binary representations of x and y, and

$$\begin{aligned} p_0(u) &= u_{n-1} \\ p_1(u) &= u_{n-1} + u_{n-2} \\ p_2(u) &= u_{n-2} + u_{n-3} \\ &\vdots \\ p_{n-1}(u) &= u_1 + u_0 \end{aligned} \tag{2.7}$$

which is obtained by using the binary representation of u. An example of binary representation is

$$(u)_{decimal} = (u_{n-1}u_{n-2}...u_1u_0)_{binary} \tag{2.8}$$

where $u_i \in \{0,1\}$. The direct application of WT to an image is faster than first creating WT matrix and then applying it.

Previously, WT is considered as an alternative to DCT and DFT since it concentrates the energy of an image to the upper left corner, and is generally used in compression and coding [30, 32–34]. Studies which use WT in face recognition differ considerably from our approach. In [31] and [35], Kekre et al. apply WT globally to face images after resizing them to the kernel size. However, it is stated that resizing an image to the kernel size decreases the success rate. In [36], PCA, DCT and WT are applied to whole image, to row mean of the image, to column mean of the image and to both row and column mean of the image. Applying the methods to whole image and to both row and column mean of the image performs better than just applying to the row mean or to the column mean. It is stated that WT is faster than the other methods. In [37], features of images are extracted by applying WT to the image and down sampling to quarter of the image. This operation is applied for seven times. For each times features of the images are extracted and compared by euclidean distance. Experiments on their own database show that applying the procedure four times and then extracting features and comparing them produce the best result. However, since the experiments are not performed on a well-known database it cannot be compared. In [38], Walsh power spectrum is used in face recognition problem. It is shown that Walsh power spectrum performs better than Fourier power spectrum and is tolerant to head movements and small illumination changes. In [39], after extracting features by synthetic discriminant functions, dimensions of features are reduced by truncated WT. All of these methods use WT globally and produce lower results.

In [40], Hassan et al. produce five images from an image by applying different formulas to the components produced by the application of WT to overlapping blocks extracted from the image. Juefei-Xu et al. use WT locally by applying WT and LBP consecutively to each pixel of periocular region of images [41–43]. In [41], the best standalone feature extraction method is explored for the periocular region of human faces for identification. It is seen that application of LBP to the features

extracted by WT performs better than other feature extraction methods such as LBP, DCT + LBP, Discrete Wavelet Transform + LBP, Force Field Transform + LBP, Gabor Filters + LBP, Laplacian of Gaussian Filters + LBP, SIFT, and SURF. Most of the kernels used in these experiments create N^2 features for each pixel of the image where $N * N$ represents size of the kernels. Application of LBP to these features reduces the N^2 features to 1, and so clusters these features for a pixel. In [42], again, WT is applied to periocular region of an image and then LBP is applied to the transformed image. After extracting features of all images in the database by this method, subspaces are built by Unsupervised Discriminant Projection (UDP) [44]. The most recent work which use a Walsh-based approach is Spartans [43]. In the feature extraction step of Spartans, various sizes of WT kernels are applied to the periocular regions of 3D face images produced from 2D images by a 3D generic elastic model. Following this, LBP is applied to the WT components, and an optimal feature subset is selected using Laplacian regularized A-optimality. Finally, they design a class-specific representation based on advanced correlation filters in kernel feature space to produce a compact representation. They also use the coupled max-pooling technique that strategically combines information from both the MACE filters and the ECPSDF filters. These methods are different than ours, and also has a lower performance.

2.3 Frontalization and Normalization Methods

In general, feature extraction methods do not perform well on images which include high pose variations. In order to solve this problem, frontalization and normalization methods are proposed. In [45], an unsupervised frontalization method called Stacked Progressive Auto-Encoders (SPAE) is proposed. This method frontalizes the images gradually and stores all stages of the frontalization. For face recognition, it uses all variations of images except the one which contains highest variation of pose. In order to increase discriminating property of the method and make it a supervised method, they apply FLD method. Experiments show that it produces better results than 2D methods which frontalize images or projects them to a common space, and outperform some 3D methods.

In [46], face images are frontalized by projecting them to a single 3D reference face image unlike most of the other frontalization methods. Thus, the method also aligns

images while frontalizing. Due to pose variations some parts of faces are invisible. The occluded parts are determined by using the reference image. After predicting the invisible parts of the face images, these are filled with values in the symmetric position. The experiments on the LFW database show that frontalizing images by this method boost performance about 3%.

In [47], a High-fidelity Pose and Expression Normalization (HPEN) method is proposed to automatically generate a natural face image in frontal pose and neutral expression. In this method, first, landmarks of images are fit to 3D model and then, the whole image is meshed into a 3D object. The pose and expression variations of images are eliminated. Lastly, the invisible regions caused by occlusion are inpainted by a method based on Possion Editing. Experiments on the Multi-PIE and LFW show that the HPEN significantly improves face recognition performance.

3. LOCAL WALSH TRANSFORM

In this chapter, first the proposed dense local image representation method Local Walsh Transform (LWT) is explained in detail. Then, the databases used during the dissertation are introduced. Finally, experimental results for the LWT method is given and complexity of the proposed method is discussed.

3.1 Local Walsh Transform (LWT)

In order to generate a face recognition system which is robust to local variations we propose a method called Local Walsh Transform (LWT) [48–50]. Applying the LWT method to an image by utilizing a N*N WT kernel, decomposes the image into N^2 images with sizes equal to those of the original image. To create these images, namely LWT components, WT kernel is applied to each pixel of the image in-turn so that for every visited pixel group, N^2 values are obtained for the N*N WT kernel. These N^2 values are assigned to the related pixel of each LWT components, which have same index as the visited pixel.

For example, by applying LWT with a 4*4 WT kernel to an image with the dimensions of 130*150, we obtain 16 LWT components. We start the application of the LWT from the (0, 0) pixel of the image. When we apply a 4*4 WT kernel to the region starting from the (0, 0) pixel, we obtain 16 values and assign these 16 values to the (0, 0) pixels of the 16 LWT components. We then pass the WT kernel to the next pixel i.e. the (0, 1) pixel and apply WT to this region. Again, we obtain 16 new values and assign these values to the (0, 1) pixels of the LWT components. This procedure is applied to each pixel of the image to obtain the pixel values of the LWT components.

To better illustrate, the LWT method can be explained as follows. While applying WT to an area, each pixel of the output is created by a different filter as can be seen in Figure 2.10. So, each LWT component can be created by applying these filters to the image separately. For example, the first filter is a summation filter which adds the values of all the pixels that it covers. The first LWT component is thereby constructed

through the application of this filter to each pixels of the image. For a 4*4 WT, a second LWT filter adds the values of the pixels in the first and second columns that it covers, and subtracts the values of the pixels in the third and fourth columns. The second LWT component is constructed by applying this filter to each pixels of the image; and so on. Thus, a LWT component $\psi_{u,v}$ on a pixel (l,m) can be expressed as

$$\psi_{u,v}(l,m) = F(u,v,l,m) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(l+x,m+y)(-1)^{q(x,y,u,v)} \tag{3.1}$$

where $u \in \{0,...,N-1\}$, $v \in \{0,...,N-1\}$, $l = 0,...,L-N$, $m = 0,...,M-N$, L and M are numbers of rows and columns of the image (f) respectively, and $q(x,y,u,v)$ is given in Equation 2.6. By computing $\psi_{u,v}$ for all (u,v) values, we generate the so-called LWT representation of an image, consisting of N^2 images, as shown by the set Ψ by

$$\Psi = \{\psi_{0,0}\,,\ \psi_{0,1}\,,\ \ldots,\ \psi_{N-1,N-1}\}. \tag{3.2}$$

Figure 3.1 shows the outputs of the application of a 4*4 LWT to a sample image. Each of the LWT component in the figure is given in the same location as the LWT filter which is shown in Figure 2.10 (b).

As one can notice, when WT is applied to the pixels which are in the last "$kernelsize-1$" rows or in the last "$kernelsize-1$" columns of the image, the WT kernel cannot cover the "*kernelsize*kernelsize*" region. To solve this problem, we extend the image by mirroring before applying the LWT method. We mirror the image from the top and left by $\lfloor \frac{kernelsize-1}{2} \rfloor$ pixels, and from down and right by $\lceil \frac{kernelsize-1}{2} \rceil$ pixels.

Based on the fact that using phase information makes the method invariant to uniform illumination changes and blurring [51], we propose an approach to create complex images from LWT components. As can be seen from Figure 2.10 (b), there is a symmetry between the LWT filters with respect to the diagonal. For example, the filter indicated by (0, 1) takes the difference in the values of the third and fourth columns from the values of the first and second columns; the filter indicated by (1, 0) takes the difference in values of the third and fourth rows from the values of the first and second rows. Using this symmetrical relationship, we propose to match the produced LWT components with respect to the diagonal to produce complex images. The LWT components which are on the upper side of the diagonal are generally created

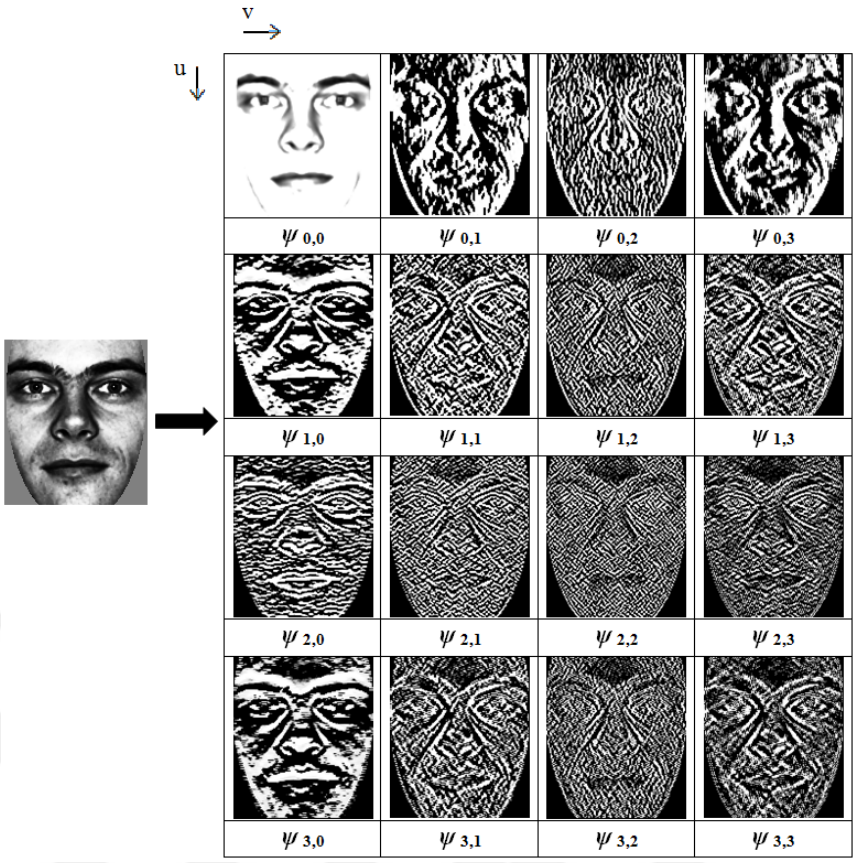


Figure 3.1 : LWT components of an image for kernel size 4*4.

by horizontal difference filters, while the LWT components which are on the lower side of the diagonal are generally created by vertical difference filters. Inspired by the shape of the Zernike polynomials which are used to produce Zernike complex components [7, 22], LWT components which are on the upper side of the diagonal are considered as real, while LWT components which are on the lower side of the diagonal are considered as imaginary parts of constituted complex images. So, if we denote the rows and columns of the WT kernel by u and v respectively, the LWT component created by the LWT filter indicated by (u,v) constitutes the real part of a complex image, and the LWT component which is created by the LWT filter indicated by (v,u) constitutes the imaginary part of the complex image where $u < v$. Formally, a LWT complex image $Z_{u,v}$ can be defined as:

$$Z_{u,v} = complex(\psi_{u,v}\,,\,\psi_{v,u}) = \psi_{u,v} + i\,\psi_{v,u} \;\;;\;\; \forall(u,v=0,...,N-1;\; u<v). \qquad (3.3)$$

For a WT order of N, the number of LWT components are N^2. By using these LWT components, the number of the produced complex images becomes $(N^2-N)/2$. For example, in Figure 3.1, there are 4 rows and 4 columns of LWT components produced

by the application of the LWT method. Through the creation of complex images, 6 complex images are built from these LWT components which are $Z_{0,1}$, $Z_{0,2}$, $Z_{0,3}$, $Z_{1,2}$, $Z_{1,3}$, and $Z_{2,3}$.

To calculate the Phase Magnitude Histogram (PMH) of a complex image, the phase and magnitude of each pixel of the complex image is calculated. The $[0, 2\pi]$ angle interval is divided into bins. The bin of the phase value of a pixel of a complex image is determined by dividing the phase value to the number of bins, and the magnitude of the pixel of the complex image is then added to this bin.

3.2 Databases

In this section, we explain the databases used during the dissertation. The proposed methods are generally evaluated in both face identification and face verification problems. We have chosen challenging, widely known, popular face databases to evaluate the methods. Specifically, for the face identification problem the Face Recognition Technology (FERET) [13] and the Surveillance Cameras Face Database (SCface) [14] databases are used, for the face verification problem the Labeled Faces in the Wild (LFW) [12] database is used.

3.2.1 The Face Recognition Technology

The Face Recognition Technology (FERET) is one of the most widely used benchmarks in the evaluation of face identification methods [13]. It contains a gallery set (Fa) of 1196 images of 1196 people and four probe sets; Fb (1195 images), Fc (194 images), Dup I (722 images), and Dup II (234 images). There is no officially determined training set of the FERET database. Some papers use an unofficially determined training set (429 images for training, 573 images for testing) which contains images from the probe sets of the FERET [21]. However, we prefer to use the gallery set as a training set, as is used for most of the state-of-the-art results [8, 52]. Since our method is an unsupervised method, not using test images during the training phase is considered more logical.

When the probe sets are compared with the gallery set, it is seen that Fb contains different facial expressions, Fc contains different illumination conditions, the images of Dup I and Dup II are taken with different cameras and have different illumination

conditions, facial expressions, occlusion, pose and scale in which the former is taken a week later, the latter is taken a year later. So, these probe sets give opportunity to evaluate methods with various facial expressions, lighting conditions, cameras, scale, and aging property. Figure 3.2 shows samples from the FERET database. The experiments are carried out by using the standard protocol of the FERET and the results are given as rank 1 accuracy rate of correct predictions [8, 13, 52].

Figure 3.2 : Samples from the FERET database.

3.2.2 The Labeled Faces in the Wild

Labeled Faces in the Wild (LFW) is a database designed to facilitate the studying of unconstrained pair matching, and is also intended to illustrate the problem of learning from one example [12]. The database consists of images collected from the web which contain variability in terms of pose, lighting, focus, resolution, facial expression, age, gender, race, accessories, make-up, occlusions, background, and photographic quality. The only constraint for these images is that the faces in the images are detected by the Viola-Jones face detector. The database contains 13233 images of 5749 different individuals. 1680 of these have more than one image. Figure 3.3 shows sample images from the LFW database.

There are six different protocols used to report performance [53]. There are two types of files called "pairs" and "people" in the database. "Pairs" files contain list of matched and unmatched pairs of people. "People" files contain name of people and number of images they include and so developers can create matched and mismatched pairs of people as they want within training or testing sets separately. For image restricted

Figure 3.3 : Samples from the LFW database.

protocol only "pairs" files are used, and for unrestricted protocol "people" files are used. The six types of standard protocol of the LFW database are:

- Unsupervised: In this protocol, the algorithm cannot access to the label information of the subjects and whether any pair of images are same or different during training and testing phases. So, it is not possible to define a threshold to determine if images are same or not. Therefore, performance has to be reported by Area Under Curve (AUC) of Receiver Operating Characteristic (ROC) curve.
- Image-restricted with no outside data: In this image restricted protocol, it is assumed that no data from outside LFW will be used including additional images, or tools such as eye-detectors, alignment methods, or feature extractors that have been trained on outside data.
- Unrestricted with no outside data: In this unrestricted protocol, it is again assumed that no data from outside LFW will be used.
- Image-restricted with label-free outside data: In this image restricted protocol using additional images from outside LFW are allowed, but these images cannot contain

the identity of any individual and cannot contain any information about whether two images are same or different.

- Unrestricted with label-free outside data: In this unrestricted protocol, again identity or labels of outside data cannot be used.
- Unrestricted with labeled outside data: This is the most permissive protocol, and allows many types of external data for training including identity of the images, arbitrary annotations, and pairs of face images labeled as same or different, as long as they do not contain individuals in an LFW test set.

There are two defined "views" of the database where View 1 is used for algorithm development and View 2 is used for performance reporting. View 1 contains training and test sets for both "pair" and "people" files. View 2 contains 10 folds of image pairs that can be used for leave-one-out cross validation. Each fold contains 300 matched and 300 unmatched pairs. In the training phase of View 2, the images of View 1 cannot be used since they may contain the same images [12, 53].

There are also four types of image sets of the LFW that one of them is original and three of the others contain aligned version of the images. The aligned images include "funneled images" [54], LFW-a images [55], and "deep funneled" images [56]. LFW-a set cannot be used in image restricted protocol.

We use the "unsupervised" protocol since during both training and testing phases we do not use the label information of the subjects, and do not access to whether any pair of images are of the same subject. For both views, we use "pairs" files and so, we use only the proposed pairs of people without generating any extra pairs.

3.2.3 The Surveillance Cameras Face Database

Images of the Surveillance Cameras Face Database (SCface) are taken in an uncontrolled indoor environment that mimics the real world conditions of the indoor face recognition problem [14]. The database contains 4160 images of 130 subjects, taken with five different quality surveillance video cameras under uncontrolled daytime and night vision conditions. These five surveillance cameras are installed in a room at a height of 2.25 m to take images of subjects at three different positions which are 4.20 m (distance 1), 2.60 m (distance 2), and 1.00 m (distance 3) far from the cameras.

Two of the cameras are capable of infra-red (IR) night vision mode and are used to take images of subjects when the room is dark. There are 21 test sets which include images taken at all three distances by the five daytime cameras (cam 1-5) and two night time cameras (called as cam 6 and 7 which are night vision mode of cam 1 and cam 5, respectively). In addition to these images of 130 subjects, mug shot images are taken under conditions expected for passport images or any other personal identification document. Figure 3.5 (a) shows mugshot image of a subject, (b), (c), and (d) show images of the subject taken at distance 1, distance 2, and distance 3, respectively, by seven different cameras settings. All images in the figure are shrunk with the same ratio. Therefore, scale difference of the images in the database can be visualized by this figure.

The experiments are carried out according to the standard protocol outlined in the original paper. According to the protocol, mug shot images are used as a gallery set and images from 21 different cases are used as test sets [14]. Results are given as rank 1 accuracy rate of correct predictions. The paper also recommends to use a different training set in the training phase which contains images of different subjects, such as FERET Fa.

3.3 Experimental Results

In order to evaluate the performance of the proposed method, LWT, we conduct a series of experiments on the FERET database. The FERET images are aligned according to the eyes, masked and cropped. The size of the images are fixed to 130*150. Then, the images are normalized to have zero mean and unit variance. We determine values of the parameters of the method by utilizing the unofficially determined training set of the FERET. We select the number of bins of PMHs as 8, after trying the number of the bins of PMHs with integer divisors of 360° total phase angle. We select the number of the subregions as 15*15 after trying from 7*7 to 24*24.

In order to determine the size of LWT kernel, we apply the LWT method with the selected parameters for size of LWT kernel as 2*2, 4*4, 8*8, 16*16, and 32*32. The method is applied to both the unofficially determined training set and the probe sets of the FERET. Table 3.1 and Figure 3.4 shows comparison of the success rates of the proposed method for these sizes of the kernel. The last column of the table is the

weighted average of the probe sets. As can be seen from the table and the figure, the best success rate is obtained by 8*8 LWT kernel.

Table 3.1 : Comparison of the success rates (%) for different sizes of LWT kernels on the FERET database.

Size of LWT Kernel	Training	Fb	Fc	Dup1	Dup2	Average
2*2	83.8	88.2	79.4	60.3	55.1	75.6
4*4	88.1	91.6	92.3	69.1	63.3	81.9
8*8	**90.9**	95.2	**92.8**	**70.2**	**66.2**	**84.4**
16*16	89.7	**96.0**	92.3	66.8	62.4	83.3
32*32	87.4	93.6	82.0	57.3	55.1	77.6

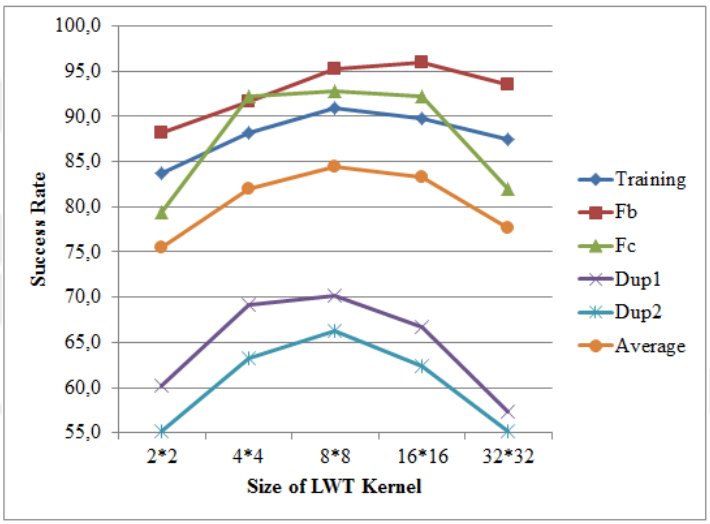


Figure 3.4 : Comparison of the success rates (%) for different sizes of LWT kernels on the FERET database.

LBP [10], LGBP [19] and LZM [22] are the most well-known local representations. In Table 3.2, we give our success rate in comparison to these methods. LWT outperforms LBP and LGBP, and is comparable to LZM. These good results of the LWT show that it is a promising method and worth to study on.

3.4 Complexity Comparison of the Methods

The complexity of the LWT method affects the complexity of other proposed LWT-based methods. So, having less complexity is important for the LWT. In this section, first we discuss complexity of the LWT. Then, we give complexity and running time of the methods in comparison to our method. Let us denote size of an image as

Table 3.2 : Success rate (%) of the LWT in comparison to other local representations on the FERET database.

Method	**Fb**	**Fc**	**Dup I**	**Dup II**	**Average**
LBP [10]	93	51	61	50	75.4
LGBP [19]	94	97	68	53	82.2
LZM [22]	95.0	87.1	73.8	70.9	85.4
LWT	**95.2**	**92.8**	**70.2**	**66.2**	**84.4**

*n*n*, number of filters as f, and size of kernel as *N*N*. We will give the complexity of methods in terms of n.

The LWT is application of LWT filters to all pixels of an image, in which values of the covered pixels are just added or subtracted. Number of the LWT filters and so number of the produced complex images depend on size of the WT kernel. In order to create real and imaginary parts of a LWT complex image, two filters are used. In our experiments we always use 8*8 WT kernel. So, number of the filters to create complex images by LWT with a *N*N* = 8*8 kernel is $f = N^2 - N = 8^2 - 8 = 56$. Application of a filter to a pixel takes $O(N^2)$ time. All of these filters contain addition or subtraction operations. The filters are applied to each pixels of an image separately. So, size of the image determines the complexity of the method. Hence, computational complexity of $T(n) = f * N^2 * n^2 = 56 * N^2 * n^2$ is $O(n^2)$.

Now, we give the complexity of the methods which are used for comparison of the LWT. The simplest method which is used for comparison is the LBP. In the LBP, the operator which takes $O(N^2)$ time for a *N*N* kernel is applied to all pixels of an image. So, the complexity of $T(n) = N^2 * n^2$ is $O(n^2)$ where *n*n* is size of the image.

In the Gabor method, filters are applied to each pixels of an image separately by convolution. According to [57], this convolution operation takes $O(N^2)$ time where *N*N* represents size of the Gabor filter. So, the complexity of application of a Gabor filter to an image is $O(N^2 * n^2)$ where *n*n* represents size of the image. In most of the papers such as [19] and [21], 40 Gabor filters of 5 scales and 8 orientations are used. So, the complexity of application of f = 40 Gabor filters is $T(n) = f * N^2 * n^2 = 40 * N^2 * n^2 = O(n^2)$.

In the LZM, same approach is used with the Gabor. The LZM filters are applied to an image pixel by pixel. The filters are created once in the beginning to reduce time

complexity [22]. Then, they are applied to each pixels of the image by convolution. If f, N*N, and n*n represent number of the filters, size of the kernel, and size of an image respectively, $T(n)$ is equal to $f * N^2 * n^2$. In LZM, number of the LZM filters are generally selected as 12. So, the complexity of $T(n) = f * N^2 * n^2 = 12 * N^2 * n^2$ is $O(n^2)$.

To sum up, general approach in the application of the methods is first creating filters and then applying these filters to each pixels of an image. So, complexity of each method converges to $O(n^2)$. Here, the simplest method is the LBP since it uses just one filter with a low complexity. The second simplest method is our LWT method, since it just adds or subtracts the values of the pixels that it covers. The running time complexities of the Gabor and the LZM are greater than that of the LWT, since they require convolution operation with floating numbers. In addition, size of the Gabor filters are bigger than ours. Specifically, computation time of application of basic versions of these methods on a computer with Intel Core i7 2.79 GHz and 8 GB RAM is given in Table 3.3 in terms of average running time (ms) versus frame per second for about 10000 run. Standard size of kernels are used for the methods which are 3*3 for LBP[1], 5*5 and 7*7 for LZM[2], 40*40 for Gabor[3], and 8*8 for LWT. Computation times of the methods are as expected that the LBP has the lowest, the LWT has the second lowest, and the Gabor has the highest running times.

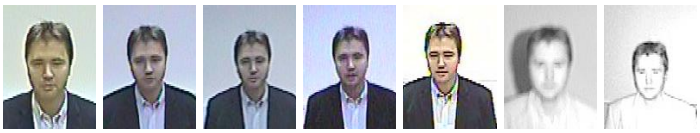
Table 3.3 : Computation time (Average running time (ms) / Frame per second (fps)) of the LWT in comparison to other methods.

Size of image	**LBP**	**LZM**	**Gabor**	**LWT**
40*40	0.02 / 50000	24 / 42	28 / 36	4 / 250
50*50	0.003 / 33333	36 / 28	52 / 19	7 / 143
100*100	0.1 /10000	106 / 9	164 / 6	25 / 40
150*150	0.2 / 5000	127 / 8	372 / 3	55 / 18
200*200	0.3 / 3333	139 / 7	635 / 2	93 / 11
250*250	0.6 / 1667	193 / 5	1037 / 1	148 / 7

[1]Code is taken from https://github.com/pi19404/OpenVision
[2]Code is taken from [22]
[3]Code is taken from https://github.com/Ekberjan/GaborFeatureExtactor

(a)

(b)



(c)

(d)

Figure 3.5 : Samples from SCface database. (a) Mugshot image of a subject. (b) Images of the subject taken at distance 1 by seven different cameras settings. (c) Images of the subject taken at distance 2 by seven different cameras settings. (d) Images of the subject taken at distance 3 by seven different cameras settings.

4. IMPROVEMENTS ON THE LOCAL WALSH TRANSFORM

In this chapter, we explain the LWT-based methods which are proposed to improve a face recognition system. The following sections show the proposed methods and the experiments.

4.1 Weighting

We adopt the weighting approach proposed in [10] to use distinctive parts of face images effectively. The details of this approach is explained in Section 2.1. However, the number of subregions in our study is different than the number of the subregions used in the LBP. So, we resize 7*7 weight map values to the number of subregions used in our study (i.e. 15*15 as determined in Section 3.3) by cubic interpolation. Table 4.1 shows result of two methods advanced by the weighting approach. It is seen that weighting approach increases success rates of both of the approaches. Our method is still better than the LBP method.

Table 4.1 : Comparison of success rates (%) of the methods advanced by the weighting approach on the FERET database.

Method	**Fb**	**Fc**	**Dup I**	**Dup II**	**Average**
LBP (Weighted) [10]	97	79	66	64	82.7
LWT (Weighted)	96.7	96.4	74.5	72.7	87.5

4.2 Separating into more Subregions

Partitioning complex images into subregions before calculating PMHs is useful to protect spatial information of face images [20]. However, while dividing an image into subregions, we can unintentionally divide meaningful parts of the face images into different subregions. To overcome this problem, after dividing a complex image into a*a subregions, we divide the same image into $(a-1)$*$(a-1)$ same sizes of subregions by shifting the starting point by half of the subregion size from top and left. So, we obtain $a^2+(a-1)^2$ subregions for each complex image. Thus, logically

combined parts may occur together at least once. By this approach, success rate on Fb, Fc, Dup I and Dup II increase to 97.0%, 96.4%, 74.8%, and 73.1% respectively by using same parameters determined in Section 3.3. So, this approach makes a slight increment on the accuracy rates of probe sets. It increases the success rate especially for Dup I and Dup II which contain more variations than Fb and Fc such as aging, and scaling.

4.3 Cascaded LWT

LWT filters are a type of high pass filter which eliminates the effect of illumination and allows the extraction of shape information of the images. In Cascaded LWT (CLWT) approach, we apply LWT again both to real and to imaginary parts of the LWT complex images. In other words, we once more apply LWT to sharpened images in which shape information is extracted by the first application of the LWT. This approach gives a depth to the method and increases success rate on low resolution images in which shapes and edges are blurred. To restate, the first application of the LWT eliminates less relevant features for face recognition and the second application of the LWT extracts shape information from the sharpened images.

Using the CLWT approach remarkably increases the number of the produced LWT complex images, compared to the LWT. In order to solve this problem, we select c complex images from the first and second applications of the LWT which are determined experimentally. In these circumstances, the number of the produced LWT complex images are $c + 2 * c^2$ in total.

It is also possible to apply the LWT more than twice. In other words, we may apply LWT to real and imaginary parts of the complex images produced by the CLWT. If we apply the LWT once more to images produced by the second application of the LWT and select c complex images from each application, the number of the produced LWT complex images becomes $c + 2 * c^2 + 2^2 * c^3$ in total. If we again apply once more in the same way, the number of the produced LWT complex images becomes $c + 2 * c^2 + 2^2 * c^3 + 2^3 * c^4$ in total, and so on. Experiments show that application of the LWT more than twice does not increase the success rate anymore and such an increment in the number of the LWT complex images increases the computational load.

We determine 5 LWT complex images to select both from the first and from the second application of the LWT by experiments on the unofficially determined training set of the FERET. These images are determined according to the success rate of each complex image when they are used on the training set separately. For 8*8 LWT, the selected LWT complex images are $Z_{0,2}$, $Z_{0,3}$, $Z_{1,2}$, $Z_{1,4}$, and $Z_{2,3}$. Success rate is increased to 98.1% on Fb, 97.9% on Fc, 78.0% on Dup I, and 78.6% on Dup II by using this approach with same parameters determined in Section 3.3. This approach increases the success rate especially for Dup I and Dup II.

4.4 The LWT with Block-based Dimension Reduction

By local image representation methods many features are produced, however comparison algorithms suffer from high dimension of the feature vectors. In order to solve this problem, dimension reduction methods can be used such as Principal Component Analysis (PCA) and Fisher's Linear Discriminant Analysis (FLD). Unfortunately, since dimension of feature vectors is very high but the number of samples is low applying these methods suffer from "small sample size" problem. In order to overcome this problem, Xie et al. propose Block-based FLD (BFLD) [21] as explained in Section 2.1.

FLD which is a supervised method do not produce good results when a person contain just one image in the training set [58]. PCA also has some drawbacks that it keeps low frequencies which reveal illumination and expression features, and eliminates high frequencies which contain discriminating information [8, 59]. In Whitened PCA (WPCA) method, eigenvectors are divided by square root of their eigenvalues. Through this simple intervention, the effect of eigenvectors which give information regarding details increases while the negative influence of eigenvectors with larger eigenvalues decreases.

We adopt Xie's block-based approach for dimension reduction. However, since the gallery set of our test databases contain just one image of each person, we choose WPCA method instead of FLD method. So, after extracting features of each block, we apply WPCA to reduce the dimension of features of blocks of the images. We call this new approach as Block-based WPCA (BWPCA).

In BWPCA method [48], we divide LWT complex images into non-overlapping blocks and then divide each block into non-overlapping subregions. Initially, histograms of subregions of a block are calculated. Then these histograms are concatenated to constitute histogram of a block. For each LWT complex images of an image, histogram of each block is calculated. Then, histograms of each corresponding blocks are concatenated. For example, histograms of the top left blocks of all LWT complex images of an image are concatenated to create histogram of the top left block of the image. Same operations are applied to all blocks of the LWT complex images to constitute histogram of each block of the image. This feature extraction stage is applied to all images in the training set. Then, features are represented as "image * block" matrices. WPCA dimension reduction method is applied to these matrices and projection matrices are learned. Then, features of the blocks are projected to new dimensions by these projection matrices. The projection matrices are stored to be used in testing stage. Final feature vector is created by concatenating the histograms of the projected blocks of the image.

We combine the approaches, which are introduced to improve the LWT method, in an algorithm. The training stage of the algorithm is given by Algorithm 1. Figure 4.1 illustrates the phases of the training algorithm from step 1 to step 6. Let us call these steps as feature extraction stage. Figure 4.2 illustrates steps from 7 to 9 which is processed after applying feature extraction stage to all images in the training set.

In the testing stage, feature extraction stage is applied to a test image. Dimensions of the features of the blocks of the image are reduced by related projection matrices. Then projected features are concatenated to create final feature vector. This final feature vector is compared with the final feature vectors of training images by using a comparison metric.

In the experiments, same parameters with the previous experiment are used. Number of blocks and number of subregions in a block are selected as 3*3 and 5*5, respectively. After shifting the starting point by half of the subregion size from the top and left, number of blocks and number of subregions in a block are selected as 2*2 and 7*7, respectively. Table 4.2 shows success rate of the proposed algorithm in comparison to some other methods. Some of these methods include same approaches which are used in our algorithm. It is seen that, our algorithm is better than most of the methods.

Algorithm 1 Training Stage of the LWT with Block-based WPCA

Input: Images in the training set
Output: Features of the images

for all images in the training set **do**
1. Apply LWT or CLWT to create LWT complex images.
2. Divide LWT complex images into non-overlapping subregions and blocks.
3. Divide LWT complex images into non-overlapping blocks and same sizes of subregions by shifting the starting point by half of the subregion size from top and left.
for all blocks **do**
4. Calculate PMHs of all subregions and multiply them with the weight of the subregion.
5. Concatenate histograms of subregions of the block.
end for
6. Concatenate histograms of corresponding blocks of the LWT complex images of the image.
end for
for all blocks **do**
7. Create "images * block" matrix.
8. Apply WPCA to "images * block" matrix and project it to new dimensions. Store the projection matrix.
end for
for all images **do**
9. Concatenate all the projected blocks of the image to generate final feature vector of the image.
end for

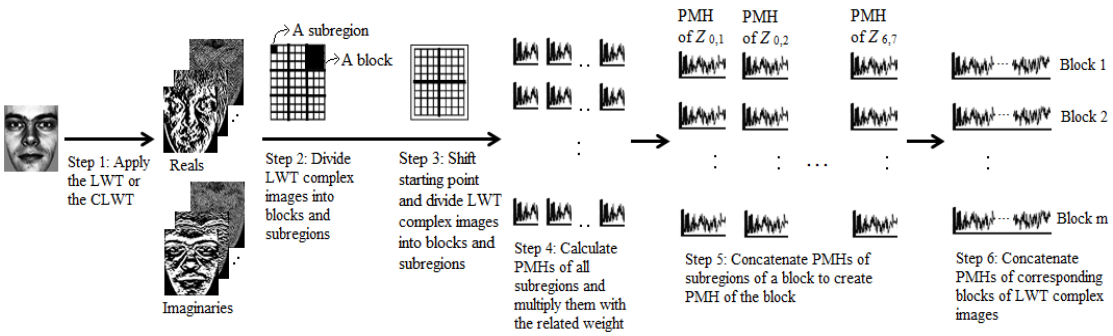


Figure 4.1 : Feature extraction of the LWT with BWPCA.

CLWT approach increases the success rate significantly especially in test sets which contain more variations.

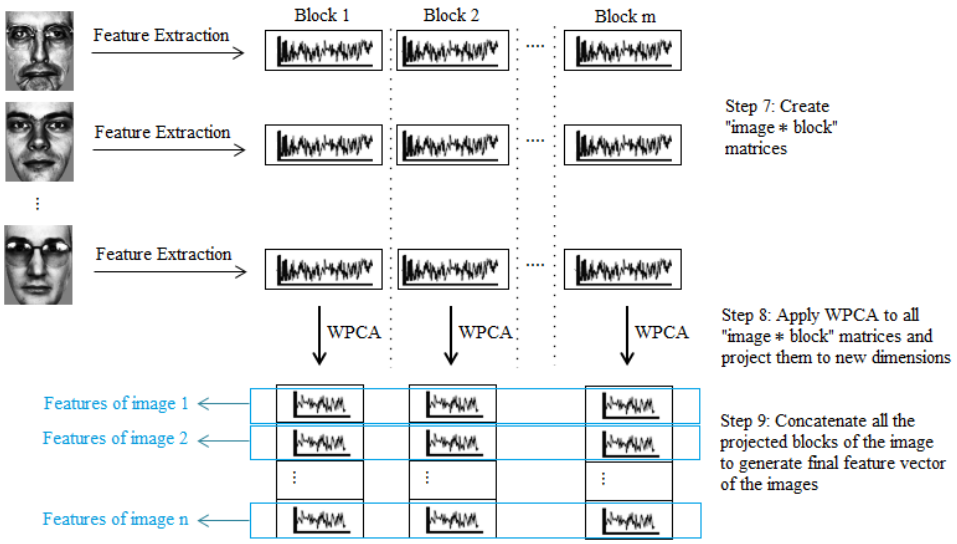


Figure 4.2 : The LWT with BWPCA method.

Table 4.2 : Success rates (%) of the LWT and the CLWT with BWPCA in comparison to other state-of-the-art methods on the FERET database.

Method	**Fb**	**Fc**	**Dup I**	**Dup II**	**Average**
LBP (Weighted) [10]	97	79	66	64	82.7
LGBP (Weighted) [19]	98	97	74	71	87.8
HGPP (Weighted) [20]	97.5	99.5	79.5	77.8	90.2
LZM^2-LPMH-IR (W) [22]	98.7	99.5	84.8	82.5	92.9
LGBP-Pha+BFLD [21]	99	99	88	86	94.3
LGBP-Mag+BFLD [21]	99	99	91	91	95.7
LGXP+BFLD [21]	99	100	92	91	96.1
The LWT with BWPCA	**99.3**	**99.0**	**82.0**	**78.2**	**91.8**
The CLWT with BWPCA	**99.5**	**100.0**	**88.8**	**88.5**	**95.1**

4.5 The LWT with XOR Approach

The details of LGXP method [21] are given in Section 2.1. In the first part of this method, after quantizing the Gabor images, XOR operator is applied to them and they are converted into grayscale images. In the same way, we apply XOR operator to the LWT complex images after quantizing them and create grayscale images. The details of the algorithm is given in Algorithm 2.

This algorithm is also improved by the approaches used in Algorithm 1. So, we enhance the algorithm by dividing images into subregions and weighting them. Figure 4.3 shows the LWT method improved by the XOR approach. Parameters are

Algorithm 2 The LWT with XOR Approach

Input: Images in the training set
Output: The number of correctly predicted test images

for all images in the training set **do**
1. Apply LWT or CLWT to create LWT complex images.
for all complex image **do**
2. Calculate phase value of each pixel by using complex images. Create a phase image with these phase values.
3. Quantize the phase image where $[0° - 90°)$ correspond to 0, $[90° - 180°)$ correspond to 1, $[180° - 270°)$ correspond to 2, $[270° - 360°)$ correspond to 3.
4. Apply XOR method to quantized image and produce grayscale image.
5. Calculate histogram of this gray level image and so create feature vector of this image. Then save the feature vector.
end for
end for
for all images in the test set **do**
6. Apply algorithm steps from 1 to 5 to the test image and so create feature vector of it.
7. Compare feature vector of the test image with feature vectors of the images in the training set. The identity of the most similar vector is determined as the identity of the test image. The predicted identity is compared with the real identity and the result is reported.
end for

selected experimentally as 8*8 for the LWT kernel size, 16*16 for subregion size, and 256 for gray level histogram bin number. Success rates of LWT-based methods improved by XOR approach used with the selected parameters are shown in Table 4.3.

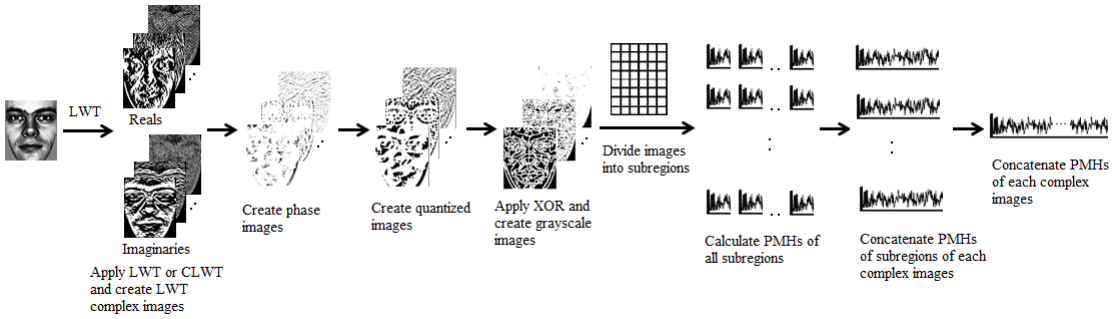


Figure 4.3 : The LWT with XOR approach.

After improving the method with these approaches, we also apply our proposed dimension reduction method BWPCA. After adding this approach to the algorithm the parameters are adapted as 15*15 for subregion size and 64 for gray level histogram bin number. Number of blocks in an image is selected as 3*3. After shifting the

Table 4.3 : Success rates (%) obtained after improving the LWT with XOR Approach on the FERET database.

Method	Fb	Fc	Dup I	Dup II	Average
LWT+XOR	95.4	93.8	65.2	55.6	82.0
LWT+XOR+Weighted	97.8	96.4	69.0	60.3	85.1
LWT+XOR+Weighted+SlidedGrid	98.0	95.9	69.3	60.7	85.2
CLWT+XOR+Weighted+SlidedGrid	97.5	97.9	78.0	72.2	89.0

starting point by half of the region size number of the blocks is selected as 2*2. According to these parameters the obtained results are shown in Table 4.4. Improving the Algorithm 1 by XOR method increases the success rate a few. So, we decide not to continue to experiments with this algorithm.

Table 4.4 : Success rates (%) after improving the LWT with XOR approach by BWPCA on the FERET database.

Method	Fb	Fc	Dup I	Dup II	Avg.
LWT+XOR+Weighted+SlidedGrid+BWPCA	99.3	97.9	76.9	71.8	89.6
CLWT+XOR+Weighted+SlidedGrid+BWPCA	99.7	100.0	89.1	87.2	95.2

4.6 Patch-based LWT

We take the advantage of both sparse and dense local representations like [25], [26] and [27] by applying the proposed local dense representation method, LWT, to salient points of face images. Patch-based LWT (PLWT) is the application of the LWT to patches which are extracted around selected landmarks of the face images and then reducing dimensions of the features of each patch. In [26], it is recommended to create five scales of images and extract same sizes of patches from each scales of the images to allow extraction of both the local and global features of the images. Taking this recommendation into account, we scale images for once or twice before applying the procedure. It has been asserted that a single global histogram of an image loses the structural information of an object [10, 20]. Therefore, dividing the image into subregions and then calculating the histogram of each subregion separately is recommended. Although this work does not use a single global image, the created patches are still large to emphasize any characteristic features of a face such as moles and scars. So, before calculating PMH of complex images, we divide them into subregions.

The feature extraction stage of the PLWT method is given in Algorithm 3. In this algorithm, first, landmarks of images are detected and some of them are selected. Then, the images and their landmarks are scaled for once or twice. For each selected landmark at different scales of an image, same size of patches are created around the landmarks. Following this, the LWT method is applied to all patches of the image and LWT complex images are created. Each complex image is divided into non-overlapping subregions and PMHs of each subregion are calculated. The PMHs of the subregions are concatenated to create the features of complex images and the features of these complex images are concatenated to create the features of the patches. So, by this algorithm, the features of the patches at all scales of an image are extracted. Figure 4.4 illustrates the feature extraction stage of the PLWT for a selected landmark.

Algorithm 3 Feature Extraction of the Patch-based LWT

Input: An image
Output: Features of patches of the image

```
Detect salient landmarks of the image.
Scale the image and the landmarks (Optional!).
for all scales of the image do
    Create same size of patches around the landmarks.
    for all patches do
        Apply LWT, and create LWT complex images.
        for all complex images do
            Divide the complex image into non-overlapping subregions.
            Calculate the PMHs of all subregions and concatenate these PMHs to create
            the features of the complex image.
        end for
        Concatenate the features of the complex images to create the features of the
        patch.
    end for
end for
```

The number of features created by Algorithm 3 is determined by the number of landmarks, scales, subregions, and bins of the PMHs, and the size of the WT kernel. The number of landmarks multiplied by the number of scales determines the number of patches. The number of features of a patch is determined by multiplying the number of LWT complex images, the number of subregions, and the number of PMH bins. The number of LWT complex images is determined by the size of the WT kernel. The dimension of the final features is the number of patches multiplied by the number of features of a patch.

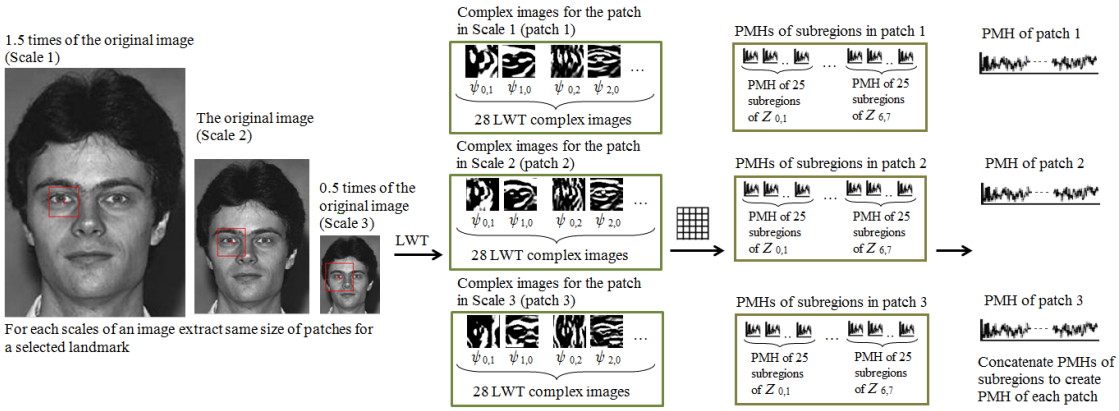


Figure 4.4 : PWLT feature extraction stage for a selected landmark. (Parameters used for this figure: Three scales of image, patch size = 50*50, LWT kernel size = 8*8, number of subregions that LWT complex features are divided into = 5*5.)

Since using such high dimensional features is a burden for the comparison metrics used to identify a test subject, the application of a dimension reduction method to the features of the patches becomes necessary. For this work, we choose an unsupervised dimension reduction method to make the proposed methods completely unsupervised and be operable on the databases which do not contain class labels of images. We choose WPCA method for the reasons explained in Section 4.4.

In the training stage, the features of the patches at all scales of the training set images are extracted by Algorithm 3. For each patch of the training images, the WPCA method is applied to learn the projection matrix of the patch. Then, after extracting features of the gallery set by Algorithm 3, the dimensions of the features of the patches of the images are reduced by using these projection matrices. In the testing stage, Algorithm 3 is applied to a test image. The dimensions of the features of each patch of the test image are reduced by using the related projection matrices. Finally, the reduced features of the patches of the test image are compared with the reduced features of the patches of all the gallery images by using a cosine similarity metric. A summation of similarity values of the features of the patches gives the final similarity value. The identity of the image with the highest similarity score is assigned as the identity of the test image for a rank 1 score. This procedure is applied to all the images in the test set. Correct predictions are counted to calculate the accuracy rate. Figure 4.5 summarizes the Patch-based LWT (PLWT) method when the training

set and the gallery set are considered as same. The code can be downloaded from https://github.com/uzunper/PLWT.

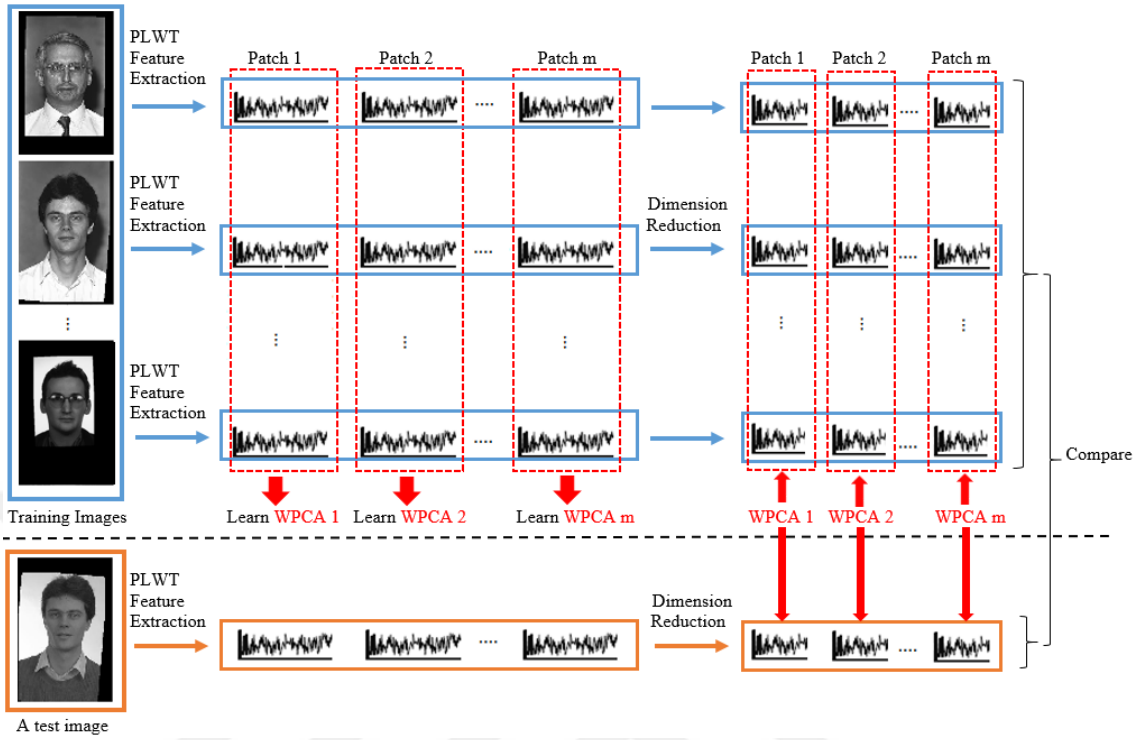


Figure 4.5 : Overview of the Patch-based LWT method.

We also use CLWT approach in Algorithm 3. To use CLWT, all that is necessary is to change the sixth line of the algorithm and apply the CLWT instead of the LWT. When the proposed method is carried out with the CLWT approach, we call this method Patch-based CLWT (PCLWT). Using the CLWT approach remarkably increases the number of the produced LWT complex images, compared to the LWT. Again, we select c complex images from the first and second applications of the LWT which are determined experimentally.

4.6.1 Landmark detection and preprocessing

The success of the landmark detection method plays a critical role in the success rate of the PLWT and the PCLWT. Therefore, the landmark detection programs are selected carefully. One of the selected successful landmark detection algorithm is Supervised Descent Method (SDM) proposed by Xiong and De La Torre [60]. The algorithm produces 49 landmarks which are 5 points for each eye brows, 4 points on nose, 5 points under the nose, 6 points for each eyes, 12 points to indicate outside edges of

lips, and 6 points for inside edges of lips. Figure 4.6 (a) shows all landmarks produced by the program on a sample image. Another successful landmark detection algorithm is in Dlib which is a C++ library containing machine learning algorithms and example codes to solve real world problems [61]. We use "Face Landmark Detection" code of this library. This code produces 68 landmarks which are 17 points around face, 5 points for each eye brows, 4 points on nose, 5 points under the nose, 6 points for each eyes, 12 points to indicate outside edges of lips, and 8 points for inside edges of lips. Figure 4.6 (b) shows all landmarks produced by this code for a sample image. These programs are not successful in all circumstances. For example, SDM may fail to detect landmarks of face images which contain occlusion and different poses, and Dlib may fail to detect landmarks in low resolution images.

(a) (b)

Figure 4.6 : Landmark points produced by (a) SDM landmark detecter (b) Dlib landmark detector.

In the preprocessing step, we apply the frontalization method of Hassner et al. [46], or the normalization method of Zhu et al. [47], or a similarity transform. The frontalization method both frontalizes and aligns the images, and the normalization method, High-Fidelity Pose and Expression Normalization (HPEN), both frontalizes images and eliminates expression variations, by using landmarks detected on the face images. Since the HPEN does not align images, similarity transform is also applied to images after the application of the HPEN.

4.6.2 Face identification experiments on the FERET database

We carry out a series of experiments on the FERET database using the standard protocol of it after applying similarity transform, the frontalization method or the HPEN as a preprocessing step. Figure 4.7 shows samples from the FERET database in which (a) shows the original images (b) shows frontalization operation applied images, c) shows images normalized by the HPEN, and (d) shows similarity transform applied and aligned images according to eyes.

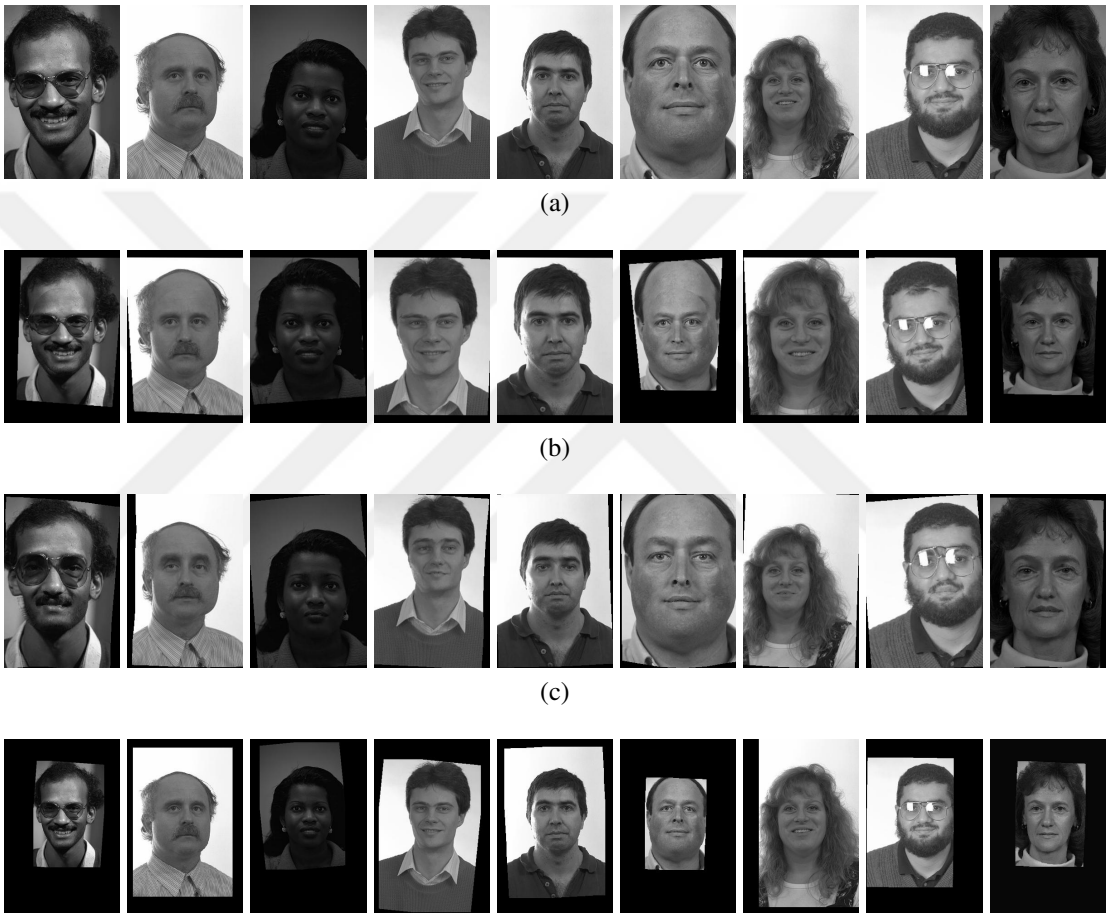


Figure 4.7 : Samples from (a) the FERET database (b) FERET images frontalized with no symmetry (c) FERET images normalized by the HPEN (d) similarity transform applied and aligned images according to eyes.

We determine the parameters of the method by utilizing the unofficial training set of the FERET database (see Section 3.2.1 for details). LWT kernel size and number of the bins of PMHs are remained same as in the previous experiments as 8*8 and 8, respectively. In order to determine the selected landmarks to be used for the PLWT and the PCLWT methods, we carry out a series of experiments on the unofficial training set of the FERET database. The landmark detection programs do not produce center

of eyes. We also designate them by using the coordinates of the landmarks at the two corners of the eyes. We select some of these landmarks and the landmarks detected by the landmark detection programs. The selected landmarks include 4, 6, 8, 10, 12, 13, 16, 18, 20, 22, 26, 49, and 51 points. We perform tests by using these landmarks for several different parameters. Figure 4.8 shows success rates of the PLWT for these different number of landmarks on the training set of the FERET by using different parameters for the size of patch and the number of subregions. These experiments show that the best performance is generally achieved by using the selected 20 landmarks for the PLWT method. Figure 4.9 shows these selected 20 landmarks on a sample image which are 6 points from eyebrows, 6 points from eyes, 4 points from nose, and 4 points from mouth. The size of patches and the number of subregions into which each patch is divided are set as 50*50 and 10*10, respectively. Figure 4.10 shows patches extracted around the selected landmarks. The dimensions of the features of the patches are reduced to one less than the number of the images in the training set. The success rate obtained with the PLWT on the frontalization method applied FERET images with the selected parameters is given in Table 4.5.

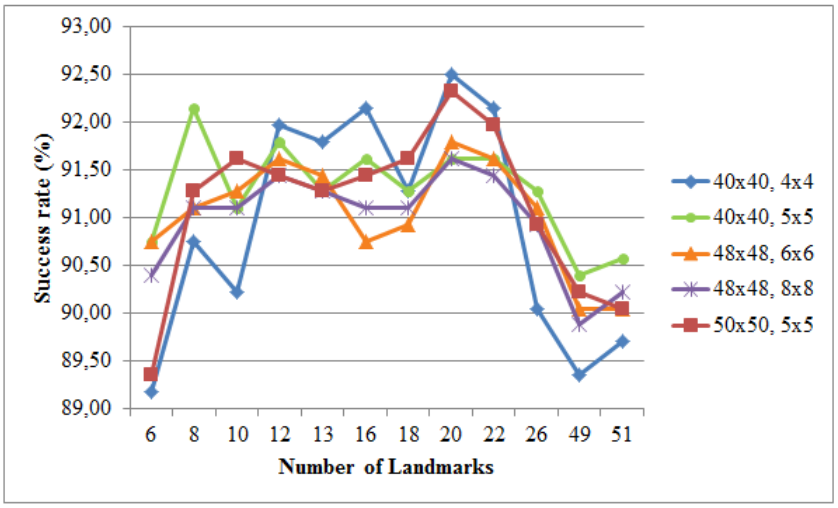


Figure 4.8 : Success rates (%) of the PLWT for different number of landmarks on the unofficial training set of the FERET database for different parameters.

For the CLWT method, five LWT complex images are chosen from the first and second application of the LWT, which are determined according to success rates of each LWT complex images when they are used on the training set separately. The selected LWT complex images for 8*8 LWT are $Z_{0,1}$, $Z_{0,2}$, $Z_{0,3}$, $Z_{0,4}$, and $Z_{1,2}$. By using the PCLWT method with these parameters, the success rate no longer increases. Choosing the LWT

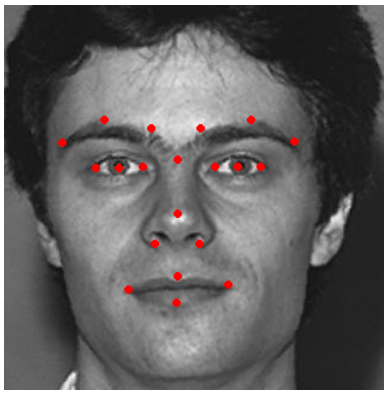
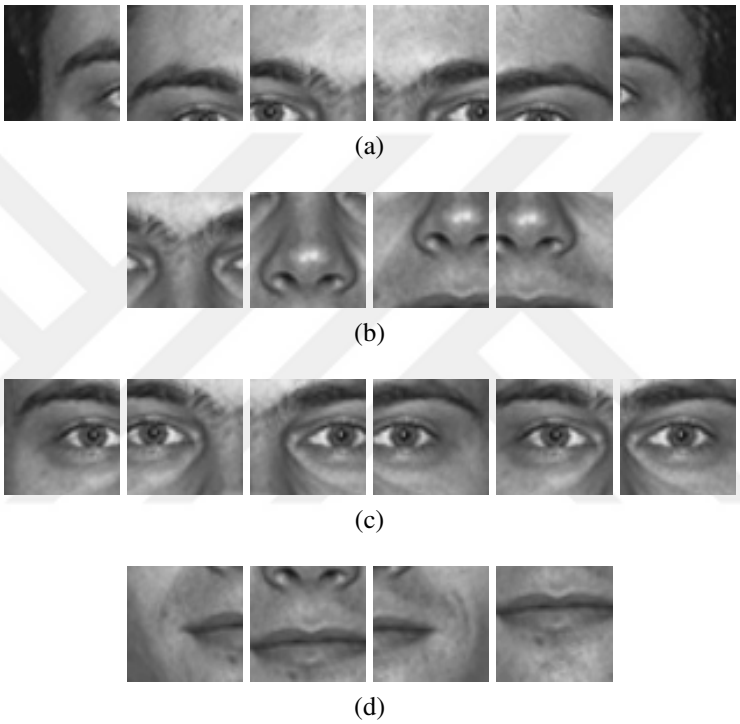


Figure 4.9 : Selected landmarks.



(a)

(b)

(c)

(d)

Figure 4.10 : Patches extracted around the selected landmarks from (a) eyebrows (b) nose (c) eyes (d) mouth.

complex images proposed in [48] and given in Section 4.3, and applying the PCLWT method with the same parameters to the images at scales of 1.0 and 1.5 times those of the originals produce the results given in Table 4.5. Although these complex images are determined by a different method, the state-of-the-art result can be obtained with these images.

The average success rate obtained with the PLWT is equal to the current published state-of-the-art result obtained by Local Patterns of Gradients (LPOG) with WPCA [52]. LPOG is built by concatenating Block-wised Elliptical LBP (BELBP), and Local Phase Quantization (LPQ) descriptions. The second best published score is obtained by

Table 4.5 : Comparison of success rates (%) of the PLWT and the PCLWT methods to the state-of-the-art results on the FERET database.

Method	**Fb**	**Fc**	**Dup I**	**Dup II**	**Avg.**
LBP [10]	97	79	66	64	82.7
LGBP [19]	98	97	74	71	87.8
HGPP [20]	97.5	99.5	79.5	77.8	90.2
LZM^{2}2-LPMH-IR (W) [22]	98.7	99.5	84.8	82.5	92.9
2DPCANet [62]	95.7	100	94.5	93.7	95.5
LGXP+BFLD [21]	99	100	92	91	96.1
G-LQP* [63]	99.9	100	93.2	91	97.0
MBC-F [23]	99.7	99.5	93.6	91.5	97.0
LMGEW//LN+LGXP [64]	99.9	100	94.7	91.9	97.5
s-POEM+POD+WPCA [8]	99.7	100	94.9	94	97.7
PCANet-2 [65]	99.6	100	95.4	94	97.8
DBN [66]	100	100	95.2	93.4	97.8
LMGEW//LN+LGBP [64]	99.9	100	95.6	93.6	98.0
SLF-RKR_l_2 [67]	99.7	99.5	96.3	94.4	98.1
Fus. of LZM-based approaches [68]	99.8	100	96	94.9	98.2
VGG-Face+WPCA [69]	99.8	99.5	96.3	96.2	98.3
LPOG WPCA [52]	99.8	100	97.4	97.0	98.8
PLWT	**99.7**	**100**	**97.4**	**97.4**	**98.8**
PCLWT	**99.7**	**100**	**98.2**	**97.9**	**99.1**

VGG-Face+WPCA [69]. In this method, VGG-face is used as a feature extractor until the 34^{th} layer. After the 34^{th} layer WPCA or LDA is applied for dimension reduction. In this table beside VGG-Face+WPCA, Deep Belief Network (DBN) [66], PCANet [65], and 2DPCANet [62] methods are also deep learning methods. The average success rate obtained with the PCLWT outperforms current published state-of-the-art method. Although applying the CLWT approach and using more than one scale of images increase the success rate by only 0.3%, we give these results to show the state-of-the-art result for the FERET.

In order to see the effect of the preprocessing step, we apply the PCLWT with the same parameters to the FERET images aligned with just similarity transform according to eye centers, and the images normalized by the HPEN and then alinged with similarity transform, and obtain the average success rates of 98.3% and 97.2% respectively. Since the PLWT and the PCLWT are robust to expression variances and most of the FERET images are already frontal, the HPEN has negative effect and the frontalization method has a small positive effect.

4.6.3 Face verification experiments on the LFW database

We carry out an excessive amount of experiments on the View 1 of the LFW database in order to adjust the parameters. First of all, we try to determine which version of the LFW database is better. For that purpose, we use the deep funneled images and LFW-a images. We have figured out that there is no superiority between these databases. Then, we try to determine the best landmark detection algorithm for the LFW database. Dlib finds landmarks better than SDM for the LFW database which contains high variances of poses and occlusion in the face images.

In the preprocessing step, the frontalization method or the HPEN with similarity transform are applied. Figure 4.11 (a) shows sample images from the LFW-a database, (b) shows these images after the application of the frontalization method [46], and (c) shows LFW images normalized by the HPEN [47].

Figure 4.11 : Samples from (a) the LFW-a database (b) LFW-a images frontalized with no symmetry (c) LFW images normalized by the HPEN.

As a comparison metric, the superiority of the cosine similarity metric and L1 Norm metric is searched. As in the FERET database the cosine similarity metric is preferred due to its success. The difference of the cosine similarity metric appears when the dimension reduction method is applied. Comparing the reduced features with cosine similarity metric produces better results than the L1 Norm.

In order to see the effect of standard normalization techniques, experiments are performed with same parameters by just changing the normalization part. Results are obtained firstly without applying any normalization technique, secondly by just normalizing to have zero mean and unit variance, and lastly by applying histogram equalization method besides normalizing to have zero mean and unit variance. LFW-a database which contains gray scale aligned images are used in these experiments. No other operation is applied to these images. It is seen that there is no effect of normalizing to have zero mean and unit variance, and applying histogram equalization to images before applying the method decreases the success rate. We also search the effect of histogram equalization method on patches. Rather than applying histogram equalization method to whole image, we apply it after extracting the patches. So, when we apply histogram equalization to patches before applying the LWT method, we again see that success rate does not increase with this normalization technique.

We search the effect of the shape of the patch on the LFW database. For this purpose we apply PLWT to horizontal rectangle, vertical rectangle, and square patches separately. We just change the size of patches as a parameter, the remaining parameters are not changed. Experiments show that the highest success rate is obtained by using vertical rectangular in which the edge along y-side is longer than the edge along x-side. Specifically, the best results is obtained by using size of patch 40*80 and number of subregions 10*10. Although the best results is obtained with vertical rectangular patches, we continue our experiments with square patches since the difference is not high and we use square patches for other databases.

We apply the PLWT to just eyes, in order to see if using just eyes are sufficient for the face verification problem. The experiments show that just using the eyes is generally not sufficient to decide if the images are of the same person or not. Applying the method to the selected 20 landmarks determined by the experiments produce better results than applying it to the landmarks around just eyes.

Based upon experiments on the View 1 of the LFW, we normalize the LFW images by the HPEN and then align by similarity transform. The size of patch and the number of subregions are set to 40*40 and 5*5, respectively. The used image scales are 1.5, 1.0, and 0.5 times those of the originals. The other parameters remain same as those found in the experiments of the FERET database. The PLWT and PCLWT methods are

applied with these parameters separately, and the dimensions of the features of each patch are reduced to 1000.

We compare the average of the AUC values of the 10 folds of the View 2 as recommended in the unsupervised category of the LFW database [70]. Table 4.6 shows the AUC values and Figure 4.12 shows the ROC curves of the methods which compete in the unsupervised category of the LFW. Our both PLWT and PCLWT methods achieve the second best results. The best result in this category is obtained by the fusion of the MLBP [17], Multiscale Local Phase Quantisation (MLPQ) [71], and Multiresolution of Binarised Statistical Image Features (MBSIF) [72] descriptors via nonlinear binary class-specific kernel discriminant analysis classifier (CS-KDA) [73]. However, this fusion is more complex than our method as explained in Section 4.6.6.

Table 4.6 : Comparison of AUC values of the PLWT and the PCLWT methods to the state-of-the-art results on the unsupervised protocol of the LFW database.[1]

Method	**AUC**
SD-MATCHES [70], 125*125, funneled	0.5407
H-XS-40 [70], 81*150, funneled	0.7547
GJD-BC-100 [70], 122*225, funneled	0.7392
LARK unsupervised [74], aligned	0.7830
LHS [75], aligned	0.8107
I-LQP* [63], aligned	forthcoming
Pose Adaptive Filter (PAF) [76]	0.9405
MRF-MLBP [77]	0.8994
MRF-Fusion-CSKDA [73]	0.9894
Spartans [43]	0.9428
LBPNet [18]	0.9404
SA-BSIF, WPCA, aligned [78]	0.9318
PLWT	**0.9608**
PCLWT	**0.9688**

In this category, the third best score is achieved by Spartans [43] which is also a WT based approach. It is completely different than our approach, as previously explained in Section 2.2. The only deep funneled method in this category, LBPNet [18], has the fifth best score and extracts high-level over-complete features by applying LBP and PCA to patches in the same topology of Convolutional Neural Network. So, our hand-crafted method also surpasses this deep learning approach.

[1]These results are taken from http://vis-www.cs.umass.edu/lfw/results.html

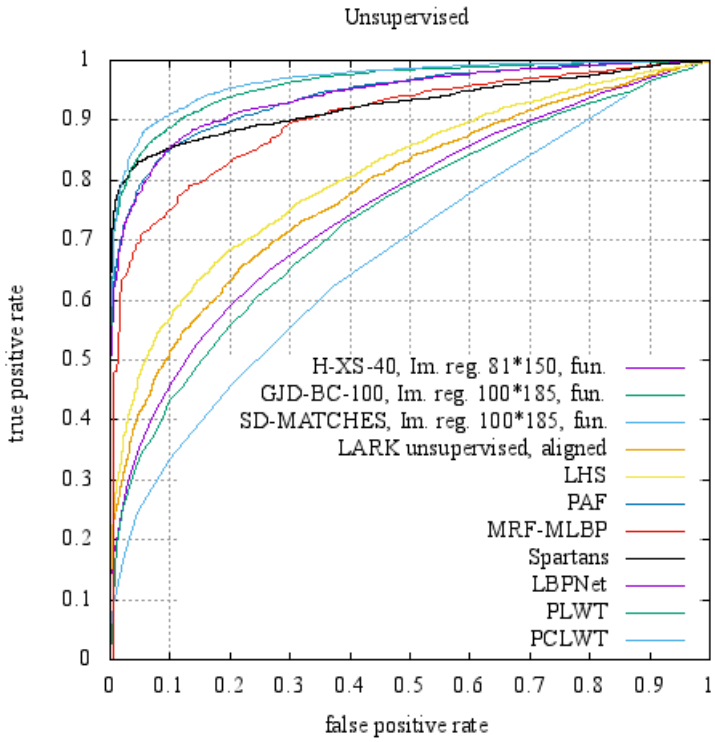


Figure 4.12 : ROC curves of different methods on View 2 of the LFW database for the "unsupervised" settings.[2]

In order to see the effect of the preprocessing step, we apply the PCLWT to the LFW-a images and the frontalization method applied LFW-a images with the same parameters, and obtain average AUC values of 0.9332 and 0.9583 respectively. It is seen that the frontalization method increases the success rate by 2.51%, and the HPEN with the similarity transform increases the success rate by 3.56% for the LFW images which contain large pose and expression variances.

4.6.4 Face identification experiments on the SCface database

After showing the superiority of the proposed methods on semi-controlled and wild images, we also validate them on low resolution images in a real life scenario. Since the SCface database contains low resolution images, both SDM and Dlib fail to detect landmarks in many images, especially in those taken 4.20 m far from the subjects. SDM cannot detect landmarks in 800 images of the 910 images taken at distance 1, whereas Dlib cannot detect any landmarks in any of these images. Taking this into account, we cannot apply the frontalization method or the HPEN to the images taken at

[2]These results and the code to plot this figure are taken from http://vis-www.cs.umass.edu/lfw/results.html

distance 1. Instead, we apply a similarity transform and align the images according to the eye coordinates given within the database. Actually, Dlib fails to detect landmarks of most of the images in this database. Since the HPEN only works with 68 landmarks we cannot apply it to this database. For the images taken at distance 2 and 3, if the landmarks can be detected by the SDM, the frontalization method is applied, otherwise similarity transform is applied according to the given eye coordinates. For each test set at different distances, all images are aligned to those dimensions i.e. for experiments at distance 1, 2, and 3, all images, including those of the FERET Fa and the mug shot images, are aligned to distance 1 dimensions (100*75), distance 2 dimensions (144*108), and distance 3 dimensions (224*168), respectively. Figure 4.13 shows the preprocessed versions of the SCface database and (a), (b), and (c) show the images at distance 1, 2, and 3, respectively. For each category, the images are arranged in the order of mug shot image, and then the images taken by cam 1 to cam 7.

(a)

(b)

(c)

Figure 4.13 : Samples from SCface database. (a) Similarity transform-applied images at distance 1. (b) Frontalized images at distance 2. (c) Frontalized images at distance 3. The first images of each category are mug shot image with the same property of the category. After mug shot images, images are taken by cam 1 to cam 7.

We avoid resizing images and fixing the size of the images to a specific dimension to prevent any deformation. However, this approach poses a problem when arranging parameters according to the size of dimensions. To solve this problem, based on the experiments on the training sets of the FERET and the LFW, we recommend choosing patch size and number of subregions of 50*50 and 10*10 respectively if the database already contains frontal images and the average distance between the eye centers is

greater than 25 pixels. For frontal high resolution images, dividing the patches into more subregions extracts the characteristics of face images well and increases the success rate. However, if the database contains high variances of poses, or the average distance between the eye centers is less than or equal to 25 pixels, it is recommended to choose a patch size and number of subregions of 40*40 and 5*5 respectively. For such images, dividing them into more subregions leads to the division and loss of characteristic information of images. So, for images taken at distance 1 and 2, patch size and number of subregions are chosen as 40*40 and 5*5, respectively; for images taken at distance 3, patch size and number of subregions are chosen as 50*50 and 10*10, respectively. Other parameters remain the same as in the other databases. As a training set, just the FERET Fa or mug shot images of the SCface database with FERET Fa are used, and the dimensions of the features of the patches are reduced to one less than the total number of images in the training set. For the images taken at distance 1 and 2, the used scales are 1.5, 1.0, and 0.5 times those of the originals; for images taken at distance 3, the used scales of images are 1.0, and 0.5 times those of the originals which are determined empirically. According to these parameters, Table 4.7, Table 4.8, and Table 4.9 show the performance of the PLWT and the PCLWT on the test images taken at distances 1, 2 and 3, respectively, and by seven different camera settings in comparison to the LPOG. The '-F' symbol in the tables symbolizes that only the FERET Fa set is used for training. It can be seen that the CLWT approach considerably increases the success rate of face recognition for low resolution images.

Table 4.7 : Success rates (%) of the PLWT and PCLWT in comparison to the LPOG on test sets at distance 1 of the SCface database.

Method	**cam1**	**cam2**	**cam3**	**cam4**	**cam5**	**cam6**	**cam7**
LPOG-F	62.3	47.7	36.2	39.2	43.1	13.1	17.7
LPOG	69.2	57.7	49.2	43.9	53.9	13.1	17.7
PLWT-F	64.6	45.4	49.2	56.9	50.8	19.2	24.6
PLWT	67.7	56.2	55.4	61.5	54.6	18.5	22.3
PCLWT-F	59.2	50.0	56.2	66.2	53.1	34.6	33.8
PCLWT	66.9	62.3	63.8	70.8	60.0	36.2	36.2

In these tables, we also compare the results with the current published state of the art result obtained by LPOG for each camera cases at each distances. It is seen that the results obtained by LPOG for cam 1 at distance 1 and cam 6 at distance 2 are greater than the results of those obtained by the PCLWT in both cases in which just FERET

Table 4.8 : Success rates (%) of the PLWT and PCLWT in comparison to the LPOG on test sets at distance 2 of the SCface database.

Method	**cam1**	**cam2**	**cam3**	**cam4**	**cam5**	**cam6**	**cam7**
LPOG-F	71.5	56.2	59.2	66.9	53.9	22.3	17.7
LPOG	73.1	66.2	63.1	75.4	52.3	23.9	20.0
PLWT-F	82.3	67.7	59.2	60.8	71.5	10.0	13.1
PLWT	82.3	66.9	57.7	59.2	73.1	10.0	14.6
PCLWT-F	89.2	80.8	70.0	70.0	86.9	16.9	25.4
PCLWT	88.5	80.8	75.4	73.1	86.2	20.0	28.5

Table 4.9 : Success rates (%) of the PLWT and PCLWT in comparison to the LPOG on test sets at distance 3 of the SCface database.

Method	**cam1**	**cam2**	**cam3**	**cam4**	**cam5**	**cam6**	**cam7**
LPOG-F	45.4	43.1	49.2	52.3	36.2	26.2	19.2
LPOG	47.7	48.5	54.6	58.5	38.5	31.5	19.2
PLWT-F	53.8	61.5	59.2	60.0	53.1	15,4	20.0
PLWT	65.4	70.0	70.8	69.2	56.9	26,9	30.0
PCLWT-F	71.5	70.8	76.9	70.0	63.1	34.6	33.1
PCLWT	73.8	76.2	81.5	76.2	66.9	46.9	36.2

Fa used as a training set or not. Other than these results, the PCLWT always surpass the current state-of-the-art results.

Table 4.10 and Table 4.11 show a comparison of the success rates of the proposed method with the state-of-the-art results which uses the DayTime and NightTime protocols proposed in [14], respectively. Some works on the SCface database which use a subset of images are not used for comparison with the results given here since they use a different protocol. Our results obtained with both the PLWT and the PCLWT surpass the state-of-the-art results in both cases in which FERET Fa, and SCface mug shot frontal images with FERET Fa are used as training sets.

For experiments with cam 6 and 7, which contain images taken in the dark, the images are deformed by frontalization operations, especially at distances 1 and 2. For these images, aligning them with a similarity transform produces better results than frontalization. Table 4.11 shows the performance on similarity transform-applied images at distances 1 and 2. In order to exemplify the bad effect of frontalization on low resolution images, Table 4.8 and Table 4.11 show the performance on frontalized images and similarity transform-applied images, respectively, taken at distance 2 by

Table 4.10 : Comparison of success rates (%) of the PCLWT with other state-of-the-art results on the SCface database using the DayTime protocol [14].

Camera/ Distance	**PCA [14]**	**Edge-preserving SR [79]**	**SR [80]**	**LBP WPCA [81]**	**ELBP(h+v) WPCA [81]**	**LPOG-F [52]**	**LPOG [52]**	**PCLWT-F**	**PCLWT**
cam1_1	2.3	11.5		43.1	43.1	62.3	69.2	59.2	66.9
cam1_2	7.7	26.9		50.0	56.2	71.5	73.1	89.2	88.5
cam1_3	5.4	11.5		41.5	45.4	45.4	47.7	71.5	73.8
cam2_1	3.1	13.9		31.5	36.9	47.7	57.7	50.0	62.3
cam2_2	7.7	21.5		44.6	50.8	56.2	66.2	80.8	80.8
cam2_3	3.9	6.2		34.6	42.3	43.1	48.5	70.8	76.2
cam3_1	1.5	10.8		20.8	34.6	36.2	49.2	56.2	63.8
cam3_2	3.9	15.4	N/A	38.5	46.9	59.2	63.1	70.0	75.4
cam3_3	7.7	10.8		49.2	51.5	49.2	54.6	76.9	81.5
cam4_1	0.7	13.1		30.0	32.3	39.2	43.9	66.2	70.8
cam4_2	3.9	24.6		50.0	50.0	66.9	75.4	70.0	73.1
cam4_3	8.5	13.1		44.6	50.8	52.3	58.5	70.0	76.2
cam5_1	1.5	16.9		28.5	36.2	43.1	53.9	53.1	60.0
cam5_2	7.7	15.4		26.9	32.3	53.9	52.3	86.9	86.2
cam5_3	5.4	11.5		23.9	31.5	36.2	38.5	63.1	66.9
Average	4.7	14.9	16.4	37.2	42.7	50.8	56.8	**68.9**	**73.5**

Table 4.11 : Comparison of success rates (%) of the PCLWT with other state-of-the-art results on the SCface database using the NightTime protocol [14].

Camera/ Distance	PCA [14]	ELBP(h+v) WPCA [81]	LPOG-F [52]	LPOG [52]	PCLWT-F	PCLWT
cam 6_1	1.5	9.2	13.1	13.1	34.6	36.2
cam 6_2	3.1	15.4	22.3	23.9	23.1	33.1
cam 6_3	3.9	25.4	26.2	31.5	34.6	46.9
cam 7_1	0.7	13.1	17.7	17.7	33.8	36.2
cam 7_2	5.4	13.1	17.7	20.0	30.0	37.7
cam 7_3	4.6	13.9	19.2	19.2	33.1	36.2
Average	3.2	15.0	19.4	20.9	**31.5**	**37.7**

cam 6 and 7. As can be seen from these tables frontalization decreases the success rate by about 10%.

4.6.5 Other experiments on face verification problem on the LFW database

The success rate of the proposed PLWT and the PCLWT methods are below the state-of-the-art results for the LFW database. The highest AUC value for the PCLWT is the second best result in the unsupervised category of the LFW. For that reason, we try to improve the method by including Support Vector Machines (SVM) and L2 Norm Metric Learning.

4.6.5.1 Experiments with Support Vector Machine

In order to decide if images are matched or unmatched, we use Support Vector Machine (SVM) method. In this method, for each pair we apply Algorithm 3 with the LWT or the CLWT to images and produce features of patches of the images. Then, we compare features of patches by cosine similarity metric. We save these similarity values in order to use later. We apply this method to all pairs in the training set. So, we obtain similarity values of patches for each pair in the training set. Number of patches are equal to the number of the selected landmarks times number of scales. Finally, we obtain a matrix, pairs versus similarity values of patches. Then we use this matrix to train SVM. In the testing stage, we again calculate similarity values of patches of the pair and then by using the trained SVM we predict if the pair is matched or unmatched.

OpenCV SVM does not run stable. Therefore, we use MATLAB to run LIBSVM which is a simple, easy-to-use library for SVM [82]. After saving similarity values of patches of the pairs in OpenCV C++, prediction operation is implemented in MATLAB. We do multi-class classification to predict if the pairs in the test set are matched or unmatched by using LIBSVM. As a kernel type of SVM we use linear, polynomial and radial basis functions. Each types of kernel uses different parameters. We perform experiments for different parameters of different kernels. Since SVM is a supervised method we use "image restricted with no outside data" protocol of the LFW and give accuracy of the test set for several parameters. Experiments are performed for size of patch and number of subregions, 50*50 and 10*10, respectively. Table 4.12 shows accuracy values of several types of kernels and several parameters for these kernels when the method is applied to experimentally selected 6 landmarks which are 43, 48, 40, 28, 23, and 21^{st} landmarks obtained by Dlib face landmark detection algorithm around eyes. The highest accuracy is obtained when polynomial kernel is applied with degree 2, gamma 100, and coef_0 0.1. The second highest accuracy is obtained with radial basis function when cost is taken as 1000 and gamma is taken as its default value $1/6$ ($1/$ *number of features*).

We repeat some of these functions with size of patch 48*48 and number of subregion 8*8. For polynomial kernel with degree 2, gamma 100, and coef_0 0.1, accuracy is 76.8%. When gamma is changed to 1, accuracy is 75.9%. For linear kernel with cost 10000, accuracy is 75.7%. For radial basis function kernel with cost 1000 and gamma 0, the accuracy is 74.3%. All of these accuracies are lower than their counterparts for size of patch 50*50 and number of subregion 10*10.

Experiments are repeated for the selected 20 landmarks. Algorithm 3 is applied to the selected 20 landmarks to extract features of patches and then similarity values of patches of image pairs are stored to be used in SVM. Again size of patches are taken as 50*50, and number of subregion 10*10 in all experiments. Table 4.13 shows accuracy values for Algorithm 3 with just LWT, just LWT applied to 1.0 and 0.5 times of images, and the CLWT for several types of kernels and parameters. Accuracies are generally higher when the method is applied to 20 landmarks rather than 6 landmarks. The CLWT makes a very slight increase. Applying more scales of images may also increase the success rate. These results are lower than most of the results compete in

Table 4.12 : Accuracy rates (%) on the LFW database when SVM is applied after extracting features by Algorithm 3 with just LWT applied on the selected six landmarks.

Kernel Type	**Cost**	**Gamma**	**Degree**	**Coef$_0$**	**Accuracy (%)**
Linear	0.01 or less				73.7
	0.1				71.5
	1				76.0
	10				76.4
	50				76.2
	100				76.1
	1000				76.5
	10000				76.6
	100000 or more				76.5
Radial Basis Function	0.01 or less	1/6			73.6
	0.1	1/6			73.6
	1	1/6			73.6
	10	1/6			76.8
	50	1/6			76.5
	100	1/6			76.4
	1000	1/6			77.1
	10000	1/6			75.2
	100000 or more	1/6			75.9
	1000	1			75.6
	1000	10			76.2
	1000	100			75.8
	1000	0.1			76.6
	1000	0.01 or less			76.4
Polynomial		0	3	0	72.6
		0	6	0	68.9
		0	2	0	73.0
		0	2	0.1	73.3
		1	2	0.1	76.3
		2	2	0.1	76.5
		100	2	0.1	77.7
		1000	2	0.1	76.5

the "image restricted no outside data" protocol of the LFW. Although this method and the PCLWT compete in different protocols, it is seen that the PLWT and the PCLWT are far more better than this supervised method.

4.6.5.2 Experiments with L2 Norm Metric Learning

In the PLWT, after applying Algorithm 3 and produce features of patches, we apply WPCA as a dimension reduction method. In this method, after application of Algorithm 3, we apply another dimension reduction method, L2 Norm Metric Learning

Table 4.13 : Accuracy rates (%) on the LFW database when SVM is applied after extracting features by Algorithm 3 with just LWT, just LWT using different scales of images, and the CLWT applied on the selected 20 landmarks.

					Accuracy (%)		
Kernel Type	**Cost**	**Gamma**	**Degree**	**Coef$_0$**	**LWT**	**LWT to Scales**	**CLWT**
Linear	0.1				74.1	75.9	
	1				77.5	77.1	76.6
	10				77.1	78.0	77.7
	100				76.4	76.0	77.0
	1000				76.3	76.2	
	10000				76.3	75.5	
Radial Basis Function	1000	0			77.4	76.6	77.1
	100	0			77.0	78.0	77.6
	1000	0.1			77.3		77.7
	1000	0.01			76.7		77.1
Polynomial		1	2	0.1	76.7		
		10	2	0.1	77.1		76.9
		100	2	0.1	77.2		77.8
		1000	2	0.1	77.0		77.4
		10000	2	0.1	76.3		

(L2ML) [83]. Algorithm 4 gives the steps of the algorithm of this dimension reduction method. In the algorithm "$eigs(S_+,k)$" function returns λ and U. Here, λ symbolizes a diagonal matrix of the k largest eigenvalues and U symbolizes the associated eigenvectors of S_+. The others steps of the algorithm consist of multiplication or division of eigenvalues and eigenvectors of $S\_$ and S_+. To adapt our method to this algorithm first we prepare the inputs of the algorithm. Input parameters are

$$S\_ = \sum_{\forall i|l_i=0} d_i d_i^T, \qquad S_+ = \sum_{\forall i|l_i=1} d_i d_i^T \tag{4.1}$$

where d_i is difference vector of features of a pair i. According to this formulation we multiply difference vector with its transpose and add all these multiplications of the pairs. Here $S\_$ is hold for unmatched pairs, S_+ is hold for matched pairs. Since we produce features of each patch separately, we produce number of patch times $S\_$ and S_+. So, we apply dimension reduction method, L2ML algorithm, to each patch separately. We store transformation matrix of each patch to use in testing stage.

In the testing stage, we again produce features of patches of images by Algorithm 3 with the LWT or the CLWT. Then, we multiply them with transformation matrices

Algorithm 4 L2 Norm Metric Learning Algorithm [83]

Input: $S_- = \sum_{\forall i|l_i=0} d_i d_i^T$, $S_+ = \sum_{\forall i|l_i=1} d_i d_i^T$
Input: Number of dimensions k of the L2ML space
Output: The L2ML transformation T

$1 : \lambda, U = eigs(S_+, k)$
$2 : U' = U\lambda^{-\frac{1}{2}}$
$3 : S'\_ = U'S\_U'^T$
$4 : \_, V = eigs(S'\_, k)$
$5 : T = VU'$

of the patches to reduce dimension. We calculate similarity of the reduced features of each patch of pairs by cosine similarity metric and add them to produce overall similarity value of the pairs. We use ROC curve to determine threshold and find the accuracy. Experiments are accomplished firstly by just using the 43^{rd} landmark obtained by Dlib face landmark detection algorithm. We apply Algorithm 3 with just LWT on 43^{rd} landmark with size of patch 50*50 and number of subregion 10*10. If we reduce the dimension of patches to 500 and 2000, the accuracy is 0.7289 and 0.7605 respectively. If we apply the algorithm to 20 landmarks with size of patch 50*50 and number of subregion 10*10 and reduce the dimension to 500, the accuracy is 0.7770. When we use 20 landmarks accuracy increases. However, this method produces less success rate than WPCA produces. There is also one drawback of the algorithm is that the size of features which is reduced by this algorithm affect the result very much. Therefore, in [83] all sizes less than the original dimension is tried for the reduced dimension to find the highest success rate.

4.6.6 Complexity comparison of the methods

In this section, we discuss the complexity of the proposed PLWT and PCLWT methods, and MRF-Fusion-CSKDA method which produce higher success rate than the PLWT and PCLWT for the LFW database. Let us represent number of filters used to produce complex images by f, size of a patch (or an image) by n*n, number of the patches by p, and number of images in the training set of a database by k. We will give complexity of the methods in terms of n and k.

In the PLWT method, the LWT is applied to patches extracted around the selected landmarks and PMHs of the produced complex images are calculated. For the PLWT

method, we select 20 landmarks and extract a constant size of patches around the landmarks. Function of running time of the application of the LWT to a patch of an image is $T(n) = f * N^2 * n^2$. To calculate PMH we reach all pixels of the complex images which means the function of running time of this operation is $T(n) = f/2 * n^2$. Therefore, the total complexity of an application of the LWT to a patch is $T(n) = f * N^2 * n^2 + f/2 * n^2 = O(n^2)$. For an image, the LWT is applied to $p = 20$ patches. So, the complexity becomes $T(n) = p * (f * N^2 + f/2) * n^2 = O(n^2)$. For k images in the training set of a database, the complexity becomes $T(n) = k * p * (f * N^2 + f/2) * n^2 = O(k * n^2)$.

In the training stage of the PLWT, after applying the method to all images in the training set of the database, PCA method is applied. The complexity of the PCA method is $O(t * d^2)$ [84] where d is the dimension of the features and t is the reduced dimension. So, time complexity of the training stage is $O(k * n^2) + O(t * d^2)$. In the PCLWT method, only number of the filters increases which do not affect the complexity. So, the complexity of training stage of the PCLWT is also $O(k * n^2) + O(t * d^2)$. In the testing stage, for a test image, after applying the LWT and reducing the dimension of the feature vector, we compare these features with reduced features of images in the training set. The complexity of comparing two vectors of t dimensions with cosine similarity is $O(t)$. If we compare the features of the test image with all images in the training set, the complexity becomes $O(t * n)$. So, the total complexity for a test image is $O(n^2 + t * n)$.

Specifically, computation time of feature extraction, dimension reduction, and comparison stages of the PLWT method for a test image for size of patch 40*40 and number of subregion 5*5 on a computer with Intel Core i7 2.79 GHz and 8 GB RAM is 190 ms, 50 ms, and 200 ms on average respectively. Before, applying the PLWT, preprocessing steps are needed. Among the preprocessing steps, Dlib and SDM landmark detection programs last 20.7 ms and 16.6 ms on average respectively; the frontalization and the HPEN last 145 ms and 965 ms on average respectively. So, for the minimum computation time, if the SDM and the frontalization method are selected as preprocessing steps, total running time of the PLWT becomes 602 ms i.e. 0.6 seconds for face identification problem. Although application of the LWT method for these parameters lasts only 4 ms, preprocessing steps, application of the

LWT to overlapping patches, and 1-NN comparison for determining the identity of the images increase computation time. For the PCLWT method, computation time with the preprocessing steps increases to 0.9 seconds, since the PCLWT creates higher dimension of features. For the face verification problem, average running time of the PLWT and the PCLWT decreases to 240 ms and 540 ms respectively for these parameters without preprocessing steps, since only one comparison is needed for decision.

Our method surpass the success rates of all compared methods except MRF-Fusion-CSKDA. Therefore, here we show the complexity of the MRF-Fusion-CSKDA is worse than that of our method. In this method, MBSIF, MLBP, MLPQ methods are applied to images which are pixel-wise aligned by a symmetric Markov Random Field (MRF) matching model. Then, CS-KDA which is proposed to solve two-class problem, is applied to the face verification problem to decide if images are matched or not. We ignore the complexity of the symmetric MRF matching model since it can be counted as pre-processing step. MBSIF is application of f filters to each pixels of multiscaled images. Then, Independent Component Analysis (ICA) dimension reduction method is applied to features of each scales of images separately. Let us show the number of the used scales by s, and size of an image by n*n. So, the complexity of MBSIF is $T(n) = k*s*f*n^2 = O(k*n^2)$ for k images in training set of the database. The complexity of ICA for each scale of images is $T(k) = 2*m*d*(d+1)*k*s = O(m*d^2*k)$ where m represents number of iterations and d represents dimension of features [85]. So, the total complexity of MBSIF is $O(k*n^2 + k*m*d^2)$.

In MRF-Fusion-CSKDA method, the second method which is used in fusion is MLBP. MLBP [17] is the application of the LBP for different radius and then application of LDA. In the method, before applying LDA, PCA ($O(t*d^2)$) is applied to reduce dimension. The complexity of application of LBP for different radius to all pixels of an image is $T(n) = N^2*n^2 = O(n^2)$ for N*N kernel. The complexity of LDA is $O(k*d*x+x^3)$ where k is the number of samples, d is the number of features and $x = min(k,d)$ [86]. In the experiments k is smaller than d. So, the complexity of LDA may be taken as $O(k^2*d+k^3)$. The total complexity of the MLBP method is $O(k*n^2+t*d^2+k^2*d+k^3)$.

In LPQ method, a Fourier transform based filter is applied to four frequency points. Then SVD is applied to frequencies of real and imaginary parts of the produced four complex images. Application of a filter to an image with size of $n*n$ is $O(n^2)$ in minimum. The complexity of SVD for $n*n$ matrix is $O(n^3)$. In MLPQ approach, multi-resolution property of the MLPQ is obtained by applying different scales of filters to the image. So, the total complexity of the MLPQ is $T(n) = s*(n^2+n^3) = O(n^3)$ for an image where s represents different scales of filters. For each image in the training set of the database it becomes $O(k*n^3)$.

In MRF-Fusion-CSKDA method, after extracting the features of images by MBSIF, MLBP and MLPQ, CS-KDA method is applied for classification. In CS-KDA method, first K_m matrix is calculated by using images and features of these images. After that Cholesky decomposition is applied to K_m. In this method, the highest complexity is caused by Cholesky decomposition whose complexity is $O(k^3-k)$ where k is the number of images in the training set of a database [73]. The other steps of the algorithm are not clear but it is stated that their complexities are less than this operation. Therefore, the total complexity of the MRF-Fusion-CSKDA method is $O(k*n^2+k*m*d^2+t*d^2+k^2*d+k^3+k*n^3) = O(k^3+k*n^3)$. The complexity of MRF-Fusion-CSKDA is greater then that of the PLWT and PCLWT since $O(k^3+k*n^3) > O(k*n^2)$.

4.7 Discussions on the Experimental Results

For a better face recognition system, the proposed LWT method is improved by weighting, separating into subregions, application of cascaded, XOR, block-based dimension reduction, and patch-based approaches. Effects of each approaches are evaluated and it is seen that each of them increases the success rate more or less. Among these approaches, reducing the dimension with WPCA method increases the success rate more than the other approaches. As explained in Section 4.4, dimension reduction methods suffer from "small sample size" problem. To overcome this problem, block-based or patch-based approaches are used. Second most effective approach to increase the success rate of the LWT-based methods is the cascaded approach. It increases the success rate especially on low resolution images. Cascaded

approach gives a depth to the method and the CLWT extracts the shape information of the low resolution images better than the LWT.

Separating an image into subregions before calculating histogram of the image increases the success rate. However, after separating an image into subregions, separating the image into same sizes of subregions by shifting the starting point by half of the subregion size increases the success rate only slightly. This approach also increases the computational load since the number of subregions extracted from an image is also increased. So, considering the success rate and computational load trade-off, this approach is not recommended for the LWT-based methods. On the other hand, extracting overlapping patches from an image and then applying the method to these patches increases success rate considerably as in the PLWT.

In the experiments, effects of the preprocessing steps are also evaluated. For the LWT-based methods, it is seen that there is no effect of normalizing to have zero mean and unit variance, and applying histogram equalization to whole image or patches before applying the method has negative effect. However, since our method is not invariant to pose variances, frontalizing images before application of the methods increases success rate considerably if the landmarks of the face images are detected correctly.

In order to determine the values of the parameters, the unofficially determined training set of the FERET and View 1 of the LFW, which is used for algorithm development, are used. An excessive amount of experiments are conducted on these databases to adjust parameters. These experiments include determining size of the LWT kernel, number of PMH bin, number of the subregions that whole, blocks or patches of images are divided into, number of the blocks, size of the patches, number of the scales, importance of the LWT complex images separately and together, and importance of the landmarks on faces separately and together. Besides, effects of similarity metrics such as L1 Norm, cosine similarity, L2 Norm, histogram intersection, chi-square, correlation, and Bhattacharyya distance are evaluated for comparison of the features. Among these metrics, the highest success rate is obtained with cosine similarity and histogram intersection. The shape of the patches for the patch-based methods, and effect of just periocular region of the face images are also evaluated. It is seen that success rate of the PLWT on vertical rectangular patches are higher than that of the PLWT

on horizontal rectangular patches and slightly higher than that of the PLWT on square patches. Applying a LWT-based method just to the periocular region of the images are not enough for high success rates.

As a result, the PLWT and the PCLWT methods produce the highest success rates among the proposed methods. In these methods, applying the LWT or the CLWT to the overlapping patches, using multi-scaled images, and applying the WPCA dimension reduction method are effective in increasing the success rate. Application of the CLWT with the patch-based approach are also effective especially on the low resolution images as in the other LWT-based methods.

5. CONCLUSIONS AND FUTURE WORK

In this dissertation, we present novel dense local image representation methods, LWT and CLWT, based on WT. The LWT decomposes an image into multiple components by utilizing WT for each pixel of the image, and allows the creation of complex images by using the symmetrical relationship between the produced LWT components. The CLWT is the application of the LWT again to both real and imaginary parts of the LWT complex images. The LWT is a strong image descriptor which reveals the shape characteristics of images. In the CLWT method, features are extracted from images already sharpened by the application of the LWT. The CLWT is particularly effective on low resolution images where shape characteristics are blurred. We also compare the complexity of the proposed methods with popular feature extraction methods. It is seen that both the LWT and the CLWT have lower running time than most of the methods. Among basic feature extraction methods LBP has lower running time than our methods. However, the success rate obtained with the LWT and the CLWT is greater than the LBP.

The PLWT is a hybrid model, which benefits from the advantages of both sparse and dense local representations. It applies the LWT method to patches extracted around the landmarks of multi-scaled images and reduces the high dimension of extracted features through WPCA. The PCLWT is application of the PLWT method with the CLWT approach. We evaluate these methods on the FERET, and SCface databases which are commonly used to examine the face recognition problem, and the LFW database which is used to examine the face verification problem. Experimental results show that both the PLWT and the PCLWT outperform most of the state-of-the-art methods. To the best of our knowledge, they achieve the best performance on both the FERET and the SCface databases, and the second best unsupervised category result on the LFW database. These high performances of the proposed methods show that these methods are effective on both wild and low resolution images, and are good alternatives to unsupervised hand-crafted methods.

As a preprocessing step, the frontalization method, a similarity transform or the HPEN with a similarity transform are applied. It is seen that the HPEN increases the success rate for high resolution images which contains large pose variances. The frontalization method increases the success rate partially on high resolution images. The similarity transform is the only alignment option for low resolution images since the landmarks of them could not be detected.

Other than these methods, the LWT is enhanced with some other approaches such as dividing image into more subregions, weighting, and reducing dimension of features by block-based dimension reduction i.e. BWPCA. In addition, LWT complex images are converted to grayscale images by XOR approach. For face verification problem, SVM is used for classification, and L2ML is used for dimension reduction. However, the success rates obtained with these methods are below the state-of-the-art results.

Both the LWT and the CLWT are effective, rich image representation methods which may be used as better alternatives to LBP and Gabor due to performance and low running time trade-off. In this dissertation, we use these methods effectively and obtain the state-of-the-art results. As a future work, these methods may be used instead of LBP or Gabor in a method. In addition, the proposed methods may be applied to a different recognition problem such as any biometrics and texture recognition. Moreover, they may be used to feed a deep learning approach. The PLWT and the PCLWT may also be used in a classification problem since they are successful in both the face identification and face verification problems.

REFERENCES

[1] **Nguyen, H.V.** (2011). *Linear Subspace Methods in Face Recognition*, (Ph.D. Thesis). The University of Nottingham.

[2] **Nguyen, H.T.** (2014). *Contributions to facial feature extraction for Face recognition*, (Ph.D. Thesis). Universite de Greenoble.

[3] **Turk, M. and Pentland, A.** (1991). Face recognition using eigenfaces, *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '91)*, pp.586–591.

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

[5] **Hafed, Z.M. and Levine, M.D.** (2001). Face Recognition Using the Discrete Cosine Transform, *International Journal of Computer Vision*, *43*(3), 167–188.

[6] **Samra, A.**, **El Taweel Gad Allah, S. and Ibrahim, R.** (2003). Face recognition using wavelet transform, fast Fourier transform and discrete cosine transform, *Proceedings of 2003 IEEE 46th Midwest Symposium on Circuits and Systems*, volume 1, pp.272–275.

[7] **Singh, C.**, **Mittal, N. and Walia, E.** (2011). Face recognition using Zernike and complex Zernike moment features, *Pattern Recognition and Image Analysis*, *21*(1), 71–81.

[8] **Vu, N.** (2013). Exploring Patterns of Gradient Orientations and Magnitudes for Face Recognition, *IEEE Transactions on Information Forensics and Security*, *8*(2), 295–304.

[9] **Lowe, D.G.** (2004). Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision*, *60*(2), 91–110.

[10] **Ahonen, T.**, **Hadid, A. and Pietikinen, M.** (2004). Face Recognition with Local Binary Patterns, *Proceedings of 8th European Conference on Computer Vision (ECCV)*, volume3021, Springer, pp.469–481.

[11] **Daugman, J.G.** (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters, *Journal of the Optical Society of America A*, *2*(7), 1160–1169.

[12] **Huang, G.B.**, **Ramesh, M.**, **Berg, T. and Learned-Miller, E.** (2007). Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, **Technical Report07-49**, University of Massachusetts, Amherst.

[13] **Phillips, P.J.**, **Moon, H.**, **Rizvi, S.A. and Rauss, P.J.** (2000). The FERET Evaluation Methodology for Face-Recognition Algorithms, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(10), 1090–1104.

[14] **Grgic, M.**, **Delac, K. and Grgic, S.** (2011). SCface – Surveillance Cameras Face Database, *Multimedia Tools and Applications*, *51*(3), 863–879.

[15] **Ojala, T.**, **Pietikäinen, M. and Mäenpää, T.** (2002). Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(7), 971–987.

[16] **Tan, X. and Triggs, B.** (2010). Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions, *IEEE Transactions on Image Processing*, *19*(6), 1635–1650.

[17] **Chan, C.H.**, **Kittler, J. and Messer, K.** (2007). Multi-scale Local Binary Pattern Histograms for Face Recognition, *Proceedings of Advances in Biometrics*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp.809–818.

[18] **Xi, M.**, **Chen, L.**, **Polajnar, D. and Tong, W.** (2016). Local binary pattern network: A deep learning approach for face recognition, *Proceedings of 2016 IEEE International Conference on Image Processing (ICIP)*, pp.3224–3228.

[19] **Zhang, W.**, **Shan, S.**, **Gao, W.**, **Chen, X. and Zhang, H.** (2005). Local Gabor Binary Pattern Histogram Sequence (LGBPHS): A Novel Non-Statistical Model for Face Representation and Recognition, *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)*, IEEE Computer Society, Washington, DC, USA, pp.786–791.

[20] **Zhang, B.**, **Shan, S.**, **Chen, X. and Gao, W.** (2007). Histogram of Gabor Phase Patterns (HGPP): A Novel Object Representation Approach for Face Recognition, *IEEE Transactions on Image Processing*, *16*(1), 57–68.

[21] **Xie, S.**, **Shan, S.**, **Chen, X. and Chen, J.** (2010). Fusing Local Patterns of Gabor Magnitude and Phase for Face Recognition, *IEEE Transactions on Image Processing*, *19*(5), 1349–1361.

[22] **Sarıyanidi, E.**, **Dağli, V.**, **Tek, S.**, **Tunç, B. and Gökmen, M.** (2012). Local Zernike Moments: A new representation for face recognition, *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pp.585–588.

[23] **Yang, M.**, **Zhang, L.**, **Shiu, S.C. and Zhang, D.** (2012). Monogenic Binary Coding: An Efficient Local Feature Extraction Approach to Face Recognition, *IEEE Transactions on Information Forensics and Security*, *7*(6), 1738–1751.

[24] **Lei, Z.**, **Pietikäinen, M. and Li, S.Z.** (2014). Learning Discriminant Face Descriptor, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(2), 289–302.

[25] **Simonyan, K.**, **Parkhi, O.M.**, **Vedaldi, A. and Zisserman, A.** (2013). Fisher Vector Faces in the Wild, *Proceedings of British Machine Vision Conference (BMVC 2013), Bristol, UK*, pp.9–13.

[26] **Chen, D.**, **Cao, X.**, **Wen, F. and Sun, J.** (2013). Blessing of Dimensionality: High-Dimensional Feature and Its Efficient Compression for Face Verification, *Proceedings of 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp.3025–3032.

[27] **Li, H.**, **Hua, G.**, **Lin, Z.**, **Brandt, J. and Yang, J.** (2013). Probabilistic Elastic Matching for Pose Variant Face Verification, *Proceedings of 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.3499–3506.

[28] **Abd-Almageed, W.**, **El-Konyaly, E. and Saraya, S.** (1997). Point feature matching adopting Walsh transform, *Proceedings of SPIE*, volume3208, pp.74–84.

[29] **Weisstein, E.W.**, Walsh Function, http://mathworld.wolfram.com/WalshFunction.html, mathWorld–A Wolfram Web Resource.

[30] **Pratt, W.K.**, **Kane, J. and Andrews, H.C.** (1969). Hadamard transform image coding, *Proceedings of IEEE*, volume57/1, pp.58–68.

[31] **Kekre, H.B.**, **Sarode, T.K.**, **Natu, P.J. and Natu, S.J.** (2011). Transform based face recognition with partial and full feature vector using DCT and Walsh transform, *Proceedings of ICWET*, ACM, pp.1295–1300.

[32] **Karagodin, M.A. and Osokin, A.N.** (2002). Image compression by means of Walsh transform, *Proceedings of the 8th International Scientific and Practical Conference of Students, Post-graduates and Young Scientists*, pp.173–175.

[33] **Kekre, H.B.**, **Sarode, T. and Natu, P.** (2013). Efficient Image Compression Technique Using Full, Column And Row Transforms On Colour Image, *International Journal of Advances in Engineering & Technology*, *6*(17), 88–100.

[34] **Kanikar, P.**, **Chaturvedi, R.N.**, **Kashyap, V. and Likhite, P.** (2013). Image Steganography using DCT, DST, Haar and Walsh Transform, *International Journal of Computer Applications*, *65*(17), 34–37, foundation of Computer Science, New York, USA.

[35] **Kekre, H.B.**, **Sarode, T.K.**, **Natu, P.J. and Natu, S.J.** (2011). Performance Comparison of Face Recognition using DCT and Walsh Transform with Full and Partial Feature Vector Against KFCG VQ Algorithm, *IJCA Proceedings on International Conference and Workshop on Emerging Trends in Technology (ICWET)*, *5*, 22–29.

[36] **Kekre, H.B.**, **Thepade, S.D.**, **Athawale, A.**, **Shah, A.**, **Verlekar, P. and Shirke, S.** (2010). Image Retrieval Using DCT on Row Mean, Column Mean and Both with Image Fragmentation, *Proceedings of the International Conference and Workshop on Emerging Trends in Technology (ICWET '10)*, ACM, New York, NY, USA, pp.102–108.

[37] **Kekre, H.B.**, **Thepade, S.D. and Maloo, A.** (2011). Face Recognition using Texture Feartures Extracted form Walshlet Pyramid, *International Journal on Recent Trends in Engineering and Technology*, *05*(01), 186–190.

[38] **Jia, X. and Nixon, M.** (1994). Analysing front view face profiles for face recognition via the Walsh transform, *Pattern Recognition Letters*, *15(6)*, 551–558.

[39] **Sadykhov, R.K.**, **Samokhval, V.A. and Podenok, L.P.** (2004). Face Recognition Algorithm on the Basis of truncated Walsh-Hadamard Transform and Synthetic Discriminant Functions, *Proceedings of Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2004)*, pp.219–222.

[40] **Hassan, M.**, **Osman, I. and Yahia, M.** (2007). Walsh-Hadamard Transform for Facial Feature Extraction in Face Recognition, *International Journal of Computer and Information Science and Engineering*, *1/3*, 167–171.

[41] **Juefei-Xu, F.**, **Cha, M.**, **Heyman, J.**, **Venogopalan, S.**, **Abiantun, R. and Savvides, M.** (2010). Robust Local Binary Pattern Feature Sets for Periocular Biometric Identification, *Proceedings of IEEE 4th International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pp.1–8.

[42] **Juefei-Xu, F.**, **Luu, K.**, **Savvides, M.**, **Bui, T.D. and Suen, C.Y.** (2011). Investigating age invariant face recognition based on periocular biometrics, *Proceedings of 2011 IEEE International Joint Conference on Biometrics (IJCB)*, pp.1–7.

[43] **Juefei-Xu, F.**, **Luu, K. and Savvides, M.** (2015). Spartans: Single-Sample Periocular-Based Alignment-Robust Recognition Technique Applied to Non-Frontal Scenarios, *IEEE Transactions on Image Processing*, *24*(12), 4780–4795.

[44] **Yang, J.**, **Zhang, D.**, **Yang, J. and Niu, B.** (2007). Globally maximizing, locally minimizing: Unsupervised discriminant projection with applications to face and palm biometrics, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*, 650–664.

[45] **Kan, M.**, **Shan, S.**, **Chang, H. and Chen, X.** (2014). Stacked Progressive Auto-Encoders (SPAE) for Face Recognition Across Poses, *Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.1883–1890.

[46] **Hassner, T.**, **Harel, S.**, **Paz, E. and Enbar, R.** (2015). Effective Face Frontalization in Unconstrained Images, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, USA.

[47] **Zhu, X.**, **Lei, Z.**, **Yan, J.**, **Yi, D. and Li, S.Z.** (2015). High-fidelity Pose and Expression Normalization for face recognition in the wild, *Proceedings of Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, pp.787–796.

[48] **Uzun-Per, M. and Gökmen, M.** (2014). Face recognition with a novel image representation: Local Walsh-Hadamard Transform, *Proceedings of 2014 IEEE 5th European Workshop on Visual Information Processing (EUVIP)*, pp.1–6.

[49] **Uzun-Per, M. and Gökmen, M.** (2015). Face recognition with Local Walsh-Hadamard Transform around landmarks, *Proceedings of 2015 IEEE 23nd Signal Processing and Communications Applications Conference (SIU)*, pp.1342–1345.

[50] **Uzun-Per, M. and Gökmen, M.** (2018). Face Recognition with Patch-based Local Walsh Transform, *Signal Processing: Image Communication*, *61*, 85–96.

[51] **Ojansivu, V. and Heikkilä, J.** (2008). Blur Insensitive Texture Classification Using Local Phase Quantization, *Proceedings of ICISP*, pp.236–243.

[52] **Nguyen, H.T. and Caplier, A.** (2015). Local Patterns of Gradients for Face Recognition, *IEEE Transactions on Information Forensics and Security*, *10*(8), 1739–1751.

[53] **Huang, G.B. and Learned-Miller, E.** (2014). Labeled Faces in the Wild: Updates and New Reporting Procedures, **Technical ReportUM-CS-2014-003**, University of Massachusetts, Amherst.

[54] **Huang, G.B.**, **Jain, V. and Learned-Miller, E.** (2007). Unsupervised Joint Alignment of Complex Images, *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pp.1–8.

[55] **Wolf, L.**, **Hassner, T. and Taigman, Y.** (2011). Effective Unconstrained Face Recognition by Combining Multiple Descriptors and Learned Background Statistics, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(10), 1978–1990.

[56] **Huang, G.B.**, **Mattar, M.**, **Lee, H. and Learned-Miller, E.** (2012). Learning to Align from Scratch, *Proceedings of Advances in Neural Information Processing Systems 25*, pp.764–772.

[57] **Ilonen, J.**, **Kamarainen, J.K. and Kälviäinen, H.** (2005). Efficient computation of Gabor features, **Technical Report100**, Department of Information Technology, Lappeenranta University of Technology.

[58] **Lu, J.**, **Plataniotis, K. and Venetsanopoulos, A.** (2005). Regularization studies of linear discriminant analysis in small sample size scenarios with application to face recognition, *Pattern Recognition Letters*, *26*(2), 181 – 191.

[59] **Liu, C. and Wechsler, H.** (1999). Comparative assessment of independent component analysis (ICA) for face recognition, *Proceedings of International Conference on Audio and Video Based Biometric Person Authentication*, pp.22–24.

[60] **Xiong, X. and De la Torre Frade, F.** (2013). Supervised Descent Method and its Applications to Face Alignment, *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.532–539.

[61] **King, D.E.** (2009). Dlib-ml: A Machine Learning Toolkit, *Journal of Machine Learning Research*, *10*, 1755–1758.

[62] **Yu, D. and Wu, X.J.** (2017). 2DPCANet: a deep leaning network for face recognition, *Multimedia Tools and Applications*, 1–16.

[63] **Hussain, S.U.**, **Napoléon, T. and Jurie, F.** (2012). Face Recognition using Local Quantized Patterns, *Proceedings of the 23rd British Machine Vision Conference*, pp.1–11.

[64] **Cament, L.A.**, **Castillo, L.E.**, **Perez, J.P.**, **Galdames, F.J. and Perez, C.A.** (2014). Fusion of local normalization and Gabor entropy weighted features for face identification, *Pattern Recognition*, *47*(2), 568–577.

[65] **Chan, T.H.**, **Jia, K.**, **Gao, S.**, **Lu, J.**, **Zeng, Z. and Ma, Y.** (2015). PCANet: A Simple Deep Learning Baseline for Image Classification?, *IEEE Transactions on Image Processing*, *24*(12), 5017–5032.

[66] **Al-Waisy, A.S.**, **Qahwaji, R.**, **Ipson, S. and Al-Fahdawi, S.** (2017). A multimodal biometrie system for personal identification based on deep learning approaches, *Proceedings of 2017 Seventh International Conference on Emerging Security Technologies (EST)*, pp.163–168.

[67] **Yang, M.**, **Zhang, L.**, **Shiu, S.C. and Zhang, D.** (2013). Robust Kernel Representation With Statistical Local Features for Face Recognition, *IEEE Transactions on Neural Networks and Learning Systems*, *24*(6), 900–912.

[68] **Basaran, E. and Gokmen, M.** (2015). An Efficient Face Recognition Scheme Using Local Zernike Moments (LZM) Patterns, *Proceedings of ACCV 2014 Workshops*, Springer International Publishing, Cham, pp.710–724.

[69] **Wan, L.**, **Liu, N.**, **Huo, H. and Fang, T.** (2017). Face Recognition with Convolutional Neural Networks and subspace learning, *Proceedings of 2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, pp.228–233.

[70] **Ruiz-del Solar, J.**, **Verschae, R. and Correa, M.** (2009). Recognition of Faces in Unconstrained Environments: A Comparative Study, *EURASIP Journal on Advances in Signal Processing*, *2009*(1), 1–19.

[71] **Tahir, M.A.**, **Chan, C.H.**, **Kittler, J. and Bouridane, A.** (2011). Face recognition using multi-scale local phase quantisation and Linear Regression Classifier, *Proceedings of 2011 18th IEEE International Conference on Image Processing*, pp.765–768.

[72] **Kannala, J. and Rahtu, E.** (2012). BSIF: Binarized statistical image features, *Proceedings of 2012 21st International Conference on Pattern Recognition (ICPR)*, pp.1363–1366.

[73] **Arashloo, S.R. and Kittler, J.** (2014). Class-Specific Kernel Fusion of Multiple Descriptors for Face Verification Using Multiscale Binarised Statistical Image Features, *IEEE Transactions on Information Forensics and Security*, *9*(12), 2100–2109.

[74] **Seo, H.J. and Milanfar, P.** (2011). Face Verification Using the LARK Representation, *IEEE Transactions on Information Forensics and Security*, *6*(4), 1275–1286.

[75] **Sharma, G.**, **ul Hussain, S. and Jurie, F.** (2012). Local Higher-Order Statistics (LHS) for Texture Categorization and Facial Analysis, *Proceedings of 12th European Conference on Computer Vision (ECCV 2012)*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp.1–12.

[76] **Yi, D.**, **Lei, Z. and Li, S.Z.** (2013). Towards Pose Robust Face Recognition, *Proceedings of 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp.3539–3545.

[77] **Arashloo, S.R. and Kittler, J.** (2013). Efficient processing of MRFs for unconstrained-pose face recognition, *Proceedings of 2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pp.1–8.

[78] **Ylioinas, J.**, **Kannala, J.**, **Hadid, A. and Pietikäinen, M.** (2015). Face Recognition Using Smoothed High-Dimensional Representation, *Proceedings of Image Analysis: 19th Scandinavian Conference, SCIA 2015*, Springer International Publishing, Cham, pp.516–529.

[79] **Mandal, S.**, **Thavalengal, S. and Sao, A.K.** (2015). Explicit and implicit employment of edge-related information in super-resolving distant faces for recognition, *Pattern Analysis and Applications*, 1–18.

[80] **Peng, Y.**, **Gökberk, B.**, **Spreeuwers, L. and Veldhuis, R.** (2012). An evaluation of super-resolution for face recognition, *Proceedings of 33rd WIC Symposium on Information Theory in the Benelux*, Enschede, the Netherlands, pp.36–43.

[81] **Nguyen, H. and Caplier, A.** (2013). Elliptical Local Binary Patterns for Face Recognition, *Proceedings of ACCV 2012 International Workshops*, Springer Berlin Heidelberg, pp.85–96.

[82] **Chang, C.C. and Lin, C.J.** (2011). LIBSVM: A Library for Support Vector Machines, *ACM Transactions on Intelligent Systems and Technology*, *2*(3), 1–27.

[83] **Barreto, R.M.**, **Ren, T.I. and Cavalcanti, G.D.C.** (2012). L2-Norm metric learning applied to unconstrained face pair-matching, *Proceedings of 2012 19th IEEE International Conference on Image Processing*, pp.581–584.

[84] **Bishop, C.M.** (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[85] **Laparra, V.**, **Camps-Valls, G. and Malo, J.** (2011). Iterative Gaussianization: from ICA to Random Rotations, *IEEE Transactions on Neural Networks*, *22*(4), 537–549.

[86] **Cai, D.**, **He, X. and Han, J.** (2008). Training Linear Discriminant Analysis in Linear Time, *Proceedings of 2008 IEEE 24th International Conference on Data Engineering*, pp.209–217.

CURRICULUM VITAE

Name Surname: Meryem UZUN-PER

Place and Date of Birth: Rize, 1985

E-Mail: meryemuzunper@gmail.com

EDUCATION:

- **B.Sc.:** 2007, Istanbul Technical University, Faculty of Electrical and Electronics Engineering, Department of Computer Engineering
- **M.Sc.:** 2011, Boğaziçi University, Institute for Graduate Studies in Science and Engineering, Department of Computer Engineering

PROFESSIONAL EXPERIENCE:

- 2010-Present, Research and Teaching Assistant in Computer Engineering Department, Istanbul Technical University.
- 2007-2009, Belbim A.Ş., Istanbul World Trade Center, Yeşilköy.
- 2006-2006, Tikle Artı Technology, ITU ARI Teknokent, Maslak.

REWARDS:

- 2016-2018, TUBITAK (The Scientific and Technological Research Council of Turkey) 2211-C PhD Scholarship Program for Priority Areas.
- 2011-2012, TURKCELL Leaders of Technology Graduate Scholarship Program.
- 2007-2009, TUBITAK 2210 Graduate Study Scholarship.
- 2003-2007, Istanbul Technical University Success Scholarship for Undergraduate Students.

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Uzun-Per, M.**, and Gökmen, M., 2018. Face Recognition with Patch-based Local Walsh Transform. *Signal Processing: Image Communication*, February, vol. 61, pp. 85–96.
- **Uzun-Per, M.**, and Gökmen, M., 2017. Face Recognition with Local Walsh Transform. *Ph.D. Consortium at 4th International Management Information Systems Conference*, 17 October, Istanbul, Turkey.

- **Uzun-Per, M.**, and Gökmen, M., 2015. Face Recognition with Local Walsh-Hadamard Transform around Landmarks. *IEEE 23. Signal Processing and Communications Applications Conference*, May 16-19, Malatya, Turkey.

- **Uzun-Per, M.**, Gökmen, M., 2014. Face Recognition with a Novel Image Representation: Local Walsh-Hadamard Transform. *IEEE 5th European Workshop on Visual Information Processing*, December 10-12, Paris, France.

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- **Uzun-Per, M.**, 2012. User Representation in Social Bookmarking Websites. *IEEE International Conference on Computer Systems and Industrial Informatics*, December 18-20, Sharjah, UAE.

- **Uzun-Per, M.**, Takçı, H., and Güngör, T., 2011. The International Conference on Artificial Intelligence, July 18-21, Nevada, USA.