ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE
ENGINEERING AND TECHNOLOGY

AN EMBEDDED DESIGN AND IMPLEMENTATION OF A FACIAL EXPRESSION RECOGNITION SYSTEM

M.Sc. THESIS

Ömer SÜMER

Department of Electronics and Communication Engineering

Electronics Engineering Programme

SEPTEMBER 2014

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE
ENGINEERING AND TECHNOLOGY

AN EMBEDDED DESIGN AND IMPLEMENTATION OF A FACIAL EXPRESSION RECOGNITION SYSTEM

M.Sc. THESIS

Ömer SÜMER
(504121374)

Department of Electronics and Communication Engineering

Electronics Engineering Programme

Thesis Advisor: Prof. Dr. Ece Olcay GÜNEŞ

SEPTEMBER 2014

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

YÜZ İFADELERİNİ TANIMA SİSTEMİ GÖMÜLÜ SİSTEM TASARIM VE UYGULAMASI

YÜKSEK LİSANS TEZİ

Ömer SÜMER
(504121374)

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Elektronik Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Ece Olcay GÜNEŞ

EYLÜL 2014

Ömer SÜMER, a M.Sc. student of ITU Graduate School of Science Engineering and Technology 504121374 successfully defended the thesis entitled **"AN EMBEDDED DESIGN AND IMPLEMENTATION OF A FACIAL EXPRESSION RECOGNITION SYSTEM"**, which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Ece Olcay GÜNEŞ** Istanbul Technical University	....
Jury Members :	**Assoc. Prof. Dr. Mürvet KIRCI** Istanbul Technical University	....
	Asst. Prof. Dr. Bülent BOLAT Yıldız Technical University	....

Date of Submission : **28 August 2014**
Date of Defense : **11 September 2014**

To my family,

FOREWORD

I would like to offer my gratitude to my supervisor, Dr. Ece Olcay Güneş, who has supported, helped and encouraged me throughout this project and coursework of this Master programme. She is not only a very good and knowledgeable supervisor but also a great motivator. She has always given me her time, encouragement and technical support that I needed during my research. and it was an honour for me to have the opportunity to work with her.

And my most sincere thanks go to all my family members; especially to my mother Ayşe Sümer, my father Bünyamin Sümer for their continuous suppport and motivation. I would like to thank my friend LTJG Ümit Kaçar for his great friendship and support during my naval service and master studies.

September 2014

Ömer SÜMER
Electronics Engineer

TABLE OF CONTENTS

Page

ABBREVIATIONS

FACS	: Facial Action Coding System
LBP	: Local Binary Pattern
SVM	: Support Vector Machine
DCT	: Discrete Cosine Transform
JAFFE	: Japanese Female Facial Expression database
FPGA	: Field Programmable Gate Arrays
MSR	: Multiscale Retinex
LTV	: Logarithmic Total Variation
DoG	: Difference of Gaussian
WLD	: Weber Local Descriptor
RBF	: Radial Basis Function
OvA	: One-versus-All
AvA	: All-versus-All

LIST OF TABLES

LIST OF FIGURES

AN EMBEDDED DESIGN AND IMPLEMENTATION OF A FACIAL EXPRESSION RECOGNITION SYSTEM

SUMMARY

In social signal processing and computer vision, there has been increasing number of studies which are related with social and behavioural sciences to some extent in last years. Affective state of human has very significant potential in many application areas such as evaluating market trends, understanding the decision-making, interpreting social interactions and their underlying background, and so on. Among the agents that make our emotions understandable, the facial expressions are the most prominent and descriptive sign of a humans's affective state. This thesis presents a literature survey on the state-of-the-art of facial expression recognition, comparison of different approaches in automatic analysis of emotions, and proposes a new embedded framework for facial expression recognition problem.

Although there have been large number of studies in facial expression recognition, the number of "affective" embedded systems are fairly scarce. In this study, an efficient embedded framework is implemented on a system-on-chip (SoC) development board. Many application areas of facial expression recognition systems necessitate the mobility, and embedded platforms which have both hardware and software development tools, as well as low power consumption and increased adaptivity.

In this study, different feature extraction methods such as local binary pattern (LBP), local ternary pattern (LTP) and Gabor filters are compared using different extraction strategies and varied kernel functions and parameters in learning phase, support vector machines (SVM). In embedded framework of facial expression system, local binary patterns and support vector machines-based methodology is preferred, because of its higher accuracy and time performance. Besides OpenCV implementation on embedded linux operating system, Zynq-7000 all programmable SoC is used to measure the performance of LBP feature extraction. Our final system has capable of facial expression recognition in both static images and video sequences at 4-5 fps.

YÜZ İFADELERİNİ TANIMA SİSTEMİ GÖMÜLÜ SİSTEM TASARIM VE UYGULAMASI

ÖZET

Sosyal sinyal işleme ve bilgisayarlı görü alanında, son yıllarda bir ölçüde sosyal bilimler ve davranış bilimleriyle ilgili yapılan çok sayıda çalışma dikkat çekmektedir. Duygu analizi, pazar eğilimlerini belirleme, karar verme mekanizmalarını anlama, sosyal ilişkiler ve ardında yatan sebepleri belirleme gibi konularda önemli bir potansiye barındırmaktadır. Duygu analizinde kullanılan tanımlayıcılar arasında en kullanışlı ve öne çıkanı, yüz ifadelerinin kullanılmasıdır. Bu tezde, otomatik yüz ifadelerinin tanınması konusunda son gelişmeler ve kullanılan yöntemler üzerine bir literatür araştırması yapılmış ve bu işlemi geçekleştirecek bir gömülü sistem çerçevesi oluşturulmuştur.

Yüz ifadeleri konusunda temel yaklaşım, doğrudan durağan duyguların sınıflandırılması ya da hareket parçacıklarının sınıflandırılmasından duygulara geçiş yapılmasıdır. Bu çalışmada Ekman tarafından farklı kültür ve toplumlarda da ayırt edici özelliği ispat edilen temel duygu sınıfları kullanılmıştır. Yüz ifadelerinin sınıflandırılması temelde n sınıflı bir sınıflandırma problemidir. Yapılan literatür taraması sonucunda daha önce kullanılan yöntemler karşılaştırmalı olarak incelenmiştir. Problemin genel çerçevesi içerisinde, ön işleme, öznitelik vektörü çıkarma, sınıfladırma işlemleri uygulanır. Ön işlemede kullanılabilecek yöntemler incelenmiş ve Tan & Triggs normalizasyonu kullanılmıştır. Öznitelik vektörü çıkarma aşamasında ise, yerel ikili örüntü (YİÖ), yerel üçlü örüntü (YÜÖ) ve Gabor filtreleri yöntemleri karşılaştırmalı olarak ele alınmış olup standart veritabanları ve deneyler üzerinde performansları incelenmiştir.

Özellikle son yirmi yılda, yüz analizi çalışmalarının hız kazanmasıyla birçok veri kümesi ve stardart deney ortaya atılmıştır. Bunların birçoğu, laboratuvar kontrollü, sabit ışık altında, poz ve duruş değişimi bulunmayan veri kümesiyken, zaman içinde bu standart koşullar altında elde edilen veri ile oluşturulan sistemlerin gerçek dünya koşullarında beklenen doğruluk oranlarında çalışmadığı görülmüştür. Bu nedenle, internet ortamında belli kilit kelimelerle yapılan aramalardan döndürülen veya TV dizileri, filmler gibi multimedya kaynaklardan derlenen veri kümeleri, kullanılan yöntemlerin test edilmesi için daha gerçekçi bir ölçü sunmaktadır. Bu durum dikkate alınarak, kullanılan yöntemler her iki türden veri kümesi üzerinde de sınanmıştır.

Bu çalışmada, öznitelik vektörü olarak yerel ikili örüntü (YİÖ), yerel üçlü örüntü (YÜÖ) ve Gabor filtreleri, öğrenme aşamasında ise destekçi karar makineleri kullanılmış olup Geliştirilmiş Cohn Kanade , MMI yüz ifadeleri, JAFFE ve SFEW veri kümelerinde çeşitli deneyler yapılarak yöntemin başarısı sınanmıştır. Bunun yanında çeşitli filmlerden seçilerek oluşturulmuş SFEW veritabanı da kullanılarak sistemin başarısı nispi olarak gerçek dünya koşullarında ve ortam şartlarının değişiklik gösterdiği görüntüler üzerinde de ölçülmüştür.

Özellikle, öznitelik çıkarma aşamasında kullanılan Yerel ikili örüntü (YİÖ) ve yerel üçlü örüntü (YÜÖ) yöntemleri literatürde yüz ifadesi analizinde kullanılan diğer yöntemlere kıyasla oldukça başarılıdır. Bu başarının nedeni, ışık veya ortam değişimleri sebebiyle gerçekleşen monoton gri seviye değişimlerinin olumsuz etkisini azaltması ve hesaplama anlamında kolaylığında yatmaktadır. Hesaplama kolaylığı özellikle yüz ifadeleri analizi gömülü sistem üzerinde yapıldığında önem kazanmaktadır. Hedef platformların işlem kapasiteleri daha karmaşık yöntemler kullanıldığında öznitelik vektörü çıkarılması aşamasında zaman kaybına sebep olduğundan nihai olarak oluşturulacak sistem video üzerinde akıcı olarak çalışmamaktadır.

Diğer veri kümelerinden farklı olarak gerçek koşullara yakın nitelikteki SFEW veritabanında, yerel ikili örüntü (YİÖ) ve destekçi karar makineleri ile yedi sınıf doğruluğu %59.76 olarak elde edilmiştir. Bu noktada, yapılacak yeni çalışmalarda yöntemlerin sınanması için standart koşullarda elde edilen görüntülerin yanı sıra, gerçek ya da gerçeğe yakın koşullarda elde edilen görsel verinin kullanılması gerektiği görülmüştür.

Deneysel sonuçlara bakıldığında, yerel üçlü örüntü (YÜÖ) ve destekçi karar makineleri kullanılarak Geliştirilmiş Cohn Kanade veritabanı üzerinde öfke, mutluluk ve şaşırma ifadeleri sırasıyla %97.78, %100 ve %97.59 başarıyla sınıflandırılmıştır. Benzer şekilde, yerel ikili örüntü (YİÖ) ve Gabor filtreleri de kullanılan veritabanları üzerinde çeşitli deneylerde kullanılmıştır. Örneğin; 5 ölçek ve 7 yönde uygulanan Gabor filtresi diğer yöntemlere yakın başarı göstermesine rağmen zaman yönünden gömülü bir uygulamada kullanıma uygun olmadığı görülmüştür.

Diğer yandan, bu çalışmanın en önemli taraflarından biri, yüz ifadelerinin sınıflandırılması gibi güncel ve kullanım alanı çok geniş olan bir probleme gömülü platformlarda çözüm ortamı oluşturmasıdır. Nitekim, yüz ifadelerinin sınıflandırılması doğası gereği mobil çözüm imkanlarını gerektirmektedir. Gömülü linux sistemler, SoC platformlar ve FPGA'lar kullanılarak yapılan çalışmalar incelendiğince yüz ifadelerinin analizini konu alan oldukça az sayıda çalışma olduğu görülmektedir.

Bilgisayar ortamında yapılan deneylerin yanı sıra, yüz ifadelerinin otomatik olarak sınıflandırılması Xilinx SoC geliştirme kartında linux (Linaro Ubuntu) işletim sistemi üzerinde C++/OpenCV geliştirme ortamı kullanılarak hem statik görüntüler, hem de videolar üzerinde gerçeklenmiştir. Gömülü sistemde, daha önce incelenen yöntemler arasından geometrik ve Tan & Triggs normalizasyonu, yerel ikili örüntü (YİÖ) ve destekçi karar makineleri kullanımıştır. Gömülü sistem uygulamasında, Geliştirilmiş Cohn Kanade veritabanındaki yüz ifadesi etiketi bulunan 327 resim kullanılarak oluşturulan destekçi karar makinesi modeli kullanılmıştır. Öte yandan, gömülü sistem üzerinde yapılan örnek uygulamada da kullanılan YİÖ öznitelik vektörleri test resimleri üzerinde uygulanarak zaman performansı ölçülmüştür.

Geliştirilen örnek uygulama hem bilgisayar ortamında, hem de kullanılan gömülü sistem platformunda çalıştırılmış ve yedi sınıflı yüz ifadeleri analizi başarıyla gerçekleştirilmiştir. Özellikle mutluluk, öfke, şaşırma ve mutsuzluk sınıflarının daha başarılı şekilde sınıflandırıldığı görülmektedir.

Bu çalışmada, daha önce gömülü platformlarda gerçeklenen yüz ifadelerini tanıma sistemleri karşılaştırmalı olarak incelenmiş ve bunlardan farklı olarak kendi gömülü sistem çerçevemiz sunulmuştur. Önerilen sistem ile, durağan resimler ve hareketli videolar üzerinde yüz ifadelerinin analizi yapılabilmektedir. Xilinx SoC geliştirme kartında linux işletim sistemi çalıştırılmış ve bir C++/OpenCV uygulaması ile sistem gerçeklenmiştir. Bu uygulama ile statik görüntüler ve video üzerinde yaklaşık olarak saniyede 4-5 görüntü hızında, yüz ifadeleri tanıma işlemi gerçekleştirilmiş ve zaman performansı açısından oldukça iyi sonuçlar elde edilmiştir.

1. INTRODUCTION

Understanding facial expressions is one of the most powerful ways in evaluating the intents and purposes of humans. It is not a straightforward task in case of difficult conditions such as lighting, pose or gaze variance, and occlusion. As well as social interactions, facial expression analysis has an important potential in human computer interaction, video surveillance, security, image and video retrieval. In this study, we will evaluate current facial expression methods, and try to find a method which is more robust to difficult conditions, and lastly implement our facial expression analysis system on an embedded platform.

1.1 Purpose of Thesis

Automatic facial expression and emotion recognition have been an interesting problem for a few decades, because successful deconstruction of facial expression into relevant emotions could be very helpful in many areas. For instance, visually impaired people could be assisted in social interactions. They are not able to interact in a proper way. A system which tells them if someone is looking them, and their facial expression of their listener could help in understanding their reaction to the narration which is started by impaired person. Not only computer vision but also psychology is an implementation area for facial expression recognition. Estimating the mood of a family picture could be a computer vision study which is intertwined with psychology. Moreover, there are continuing studies to conduct automatic deception detection.

Another interesting application area is the analysis of commercials. They could be used to measure the relation between the emotional state of the consumer and brands. A control group watch a advertising media, and facial expression analysis gives an exclusive answer to the question of whether the product is really engaged with the audience or not.

In mobile and ubiquitous computing, context-aware approach is critical, and the emotional state could be evaluated as high level contextual information. Thus, they can be used to regulate the behavior of a system such as in smart environments. Lastly, game industry could be one of the most stimulating application area. In contrast to virtual reality and graphics technology, there are mobile games which use facial expressions, as motion controller (kinect) games which are based on pose estimation.

1.2 Literature Review

A literature review shows that there are two main approaches in facial expression classification. These are *static emotion analysis* and *action unit-based analysis*. Besides, the methodology could be categorized into two groups; *appearance-based* and *geometric-based methodology*. In following subsections, these approaches and methodologies are summarized.

1.2.1 Static Emotions vs. Action Units

Two different approaches in facial expression analysis can be summarised as follows:

1. **Static emotion analysis:** According to this approach, face images are taken as a whole frontal image and processed. There are six unique facial expressions: anger, disgust, fear, sadness, happy, and surprise. Each of these expressions corresponds to a different characteristics.

2. **Action-unit analysis:** Psychologist Paul Ekman and Wallace V. Friesen [10] adopted the "Facial Action Coding System" (FACS) which is first developed and named by a Swedish anatomist Carl-Herman Hjortsjö. FACS is a system which is designed for human observers to describe facial expressions in terms of micro expressions or facial action units (AUs). Action units are based on the movements of facial muscles in human face, and facial expressions are made up of different combination of AUs.

According to this approach, one way is only to classify action units. Another alternative is to map the results of AU classification into static emotions. An example of different upper and lower action unit combinations are depicted in Fig.1.1 and 1.2.

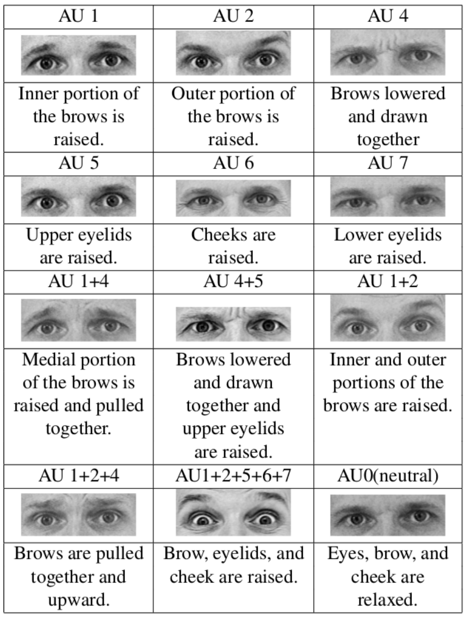

Figure 1.1: Basic upper face action units or AU combinations [3]

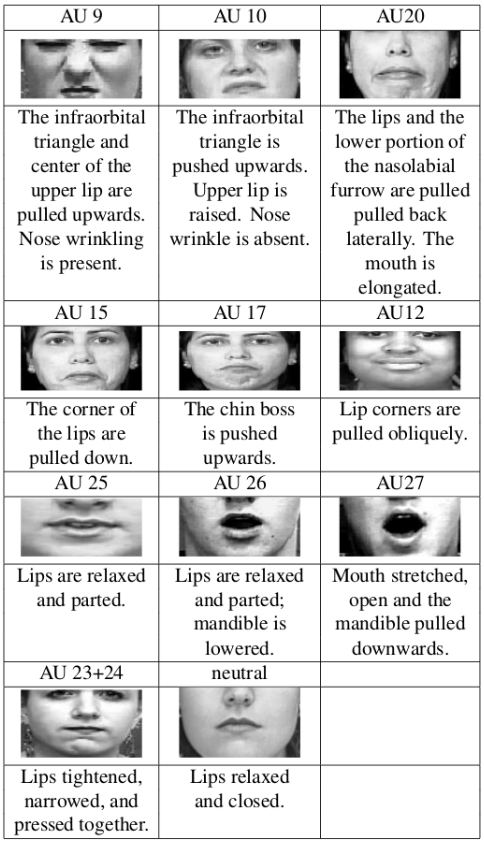

Figure 1.2: Basic lower face action units or AU combinations [3]

1.2.2 Geometric-based vs. Appearance-based Methodology

In both approaches, the state-of-the-art methodology in facial expression analsis can be divided into two main categories:

1. **Geometric-based methodology:** This methodology relies on extraction some landmarks from face images, tracking and processing the motion from these points. These methods are generally based on geometric model fitting, and use some special motion parameters or spatial points from images [11, 12] are used as a feature vector. Recently, Valstar and Pantic [13] reported that geometric representations are comparable or more discriminative than appearance-based approach. However, geometric methods need more accurate registration. In addition, it could necessitates more time and power consuming detection and tracking methods in video sequences and real-time applications. Pose and illumination changes makes difficult extraction of facial landmarks and tracking. Thus, the performance of system decrease critically. Besides, expressions can easily change in various ways and it could be difficult to predict their motion in time domain.

2. **Appearance-based methodology:** Instead of using motion or displacement of face in time, this method uses the color or intensities of image pixels. Similar to geometric-based approach, a registration or an alignment could be done, however these points are used to extract and form a feature vector. Gabor filter, Haar wavelets, discrete cosine transform (DCT) could be given as an example of appearance-based methods. Applying face images a bank of Gabor filters in different scale and orientations [14, 15] is one of the most used methods in appearance-based approach, because of its performance. Due to the computational complexity of Gabor filters, Local Binary Patterns (LBP) is used in facial analysis [16, 17]. LBP is robust to varying lighting conditions, and easy to compute.

Figure 1.3: Sample images from Cohn Kanade database [4]

In facial expression recognition, there have been a few important datasets which are used as a benchmark. An example from the first studies in automated facial expression recognition is the development of *The Japanese Female Facial Expression* (JAFFE) Database [18]. Lyons et al. proposed a method which is based on labeled elastic graph matching, 2D Gabor wavelet representation, and linear discriminant analysis and reported classification results of gender, ethnicity, and emotion [14].

Automated system which is capable of static or temporal analysis of emotions dates back to 1999. Barlett, in cooperation with Ekman and Sejnowski [19] analyze automatically facial expressions using FACS. This study leads to the development of the *Computer Expression Recognition Toolbox* (CERT) [20].

Cohn and Canade's development of a facial expression database which is action unit-labeled and contains both static images and image sequences increased the popularity of facial expression studies [4]. The Cohn-Kanade database remained as a reference and benchmark database to prove efficiency of proposed methods in facial expression recognition.

Today, the most important trend in facial expression analysis is to conduct experiments in the wild. This means that the test, training, and validation of experiments are done using realistic data, for instance from movies, tv series, or web images. Thus, we will reconsider the performance of our methods using a database which is not formed in lab controlled environment.

1.2.3 Design and Implementation

Applicability is one of the significant concerns in embedded image processing and computer vision applications, We aim at applying the facial expression analysis framework on an embedded Linux development, *Zedboard* according to performance evaluation. To implement our design in an embedded mobile platform which is short of power and computational resources, we need to take into consider various issues. Therefore, we decide which method seems best for our implementation according to the results of our performance evaluation.

1.3 Overview

This study aims at the design and implementation of a facial expression classification system on an embedded linux development board. The lack of computational and power sources bring new constraints to our design. Besides the performance evaluation of various methods in facial expression recognition, we will consider them, and choose best method for our case.

In Chapter 2 we will explain the general framework of facial expression classification, and give brief information about the methods which is used in preprocessing, feature extraction, and classification. Chapter 3 brief information about used databases and experimental setup, and finally present our performance evaluation. After that, we will describe the technical specifications of our development board (Zedboard) which is based on Xilinx: Zynq-7000 All Programmable System-on-Chips (SoCs) which establishes an integrity between field programmable gate arrays (FPGA) and ARM microprocessors in Chapter 4. Finally, we present our findings and conclusion in Chapter 5.

2. THEORETICAL BACKGROUND

In this chapter, we will explain the general framework of "facial expression classification" system and describe the methods that we used in this study. In Fig. 2.1, general framework of a facial expression system is depicted. We choose different normalization, feature extraction, and classification methods in order to make a comparative analysis of their performance using standard datasets and benchmarks.

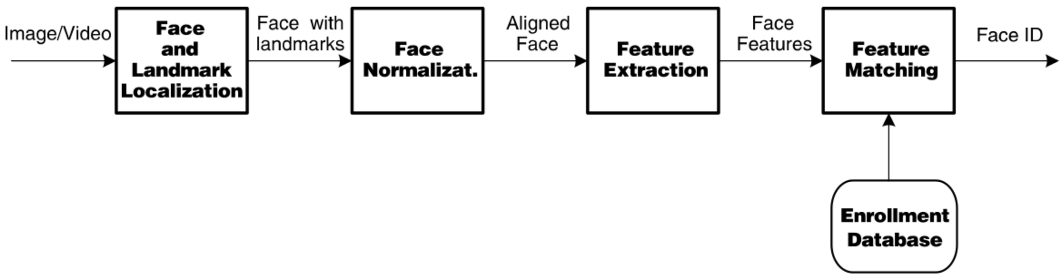

Figure 2.1: General framework of facial expression recognition system

In this chapter, we first introduce the methods that we used in image normalisation, feature extraction and classification steps, then explain the facial expression datasets and their experimental setup, and finally compare our results with state-of-the-art using same benchmarks.

2.1 Image Normalisation

In image processing and computer vision applications, the first process which is applied to input images or sequences is the *normalisation of the image* using a preprocessing method. This methods mainly depends on preferred feature extraction method. We can divide our normalization into two categories: one is the transformation which will ensure some facial points are on the same place in frames; other is the preprocessing for environmental changes such as illumination.

Looking into Fig.2.2, it can be seem how we extracted faces from input images. Normalisation is utterly important because we will use appearance-based feature

Figure 2.2: First process : *face extraction and normalisation.*

extraction methods and divide image into sub-blocks. Using the coordinates of eyes, it is a good approach to transform image and ensure the eyes in all images are on the same pixels. In this process, it will proper to check the vertical positions of eyes should be same. So, the angle between eyes can be calculated and image can be rotated using this angle. After that, the distance between eyes could also be named as *interpupilary distance (IPD)* in biometrics. In face recognition, IPD should be between 30 and 75. This resolution is necessary to detect and recognise discriminative characteristics of faces. After vertical normalization with rotating, we cropped the input image 70 % of horizontal dimension will be as eye distance, and 15% of eye distance will remains in both sides of face. Then, we applied similar process in vertical dimensions, and finally scale the image in order to ensure the dimensions of faces be same.

In preprocessing against environmental changes, some low-scale operations and filters are applied into the images in order to elude the disadvantage of the regularities which stem from typically lighting changes. Looking into the literature, there are numerous methods such as Multiscale Retinex (MSR) [21], Logarithmic Total Variation (LTV) [22], Gross & Brajovic method (GB) [23] and Tan & Triggs normalisation [24]. We will use Tan & Triggs normalisation due to higher performance in comparison to other mentioned methods.

In Tan & Triggs normalisation, first applied filter is gamma correction. It is a non-linear gray-level normalisation which imporoves the local dynamic range of image in darker regions. After gamma correction, difference of gaussian (DoG) in order to remove the gradients and shading due to illumination differences. Then, contrast equalisation is applied into the image. Alternatively, a similar method, *zero mean unit variance normalisation* can be applied in place of contrast equalization. In reference paper, an elliptical mask is applied into image after DoG, however we will not apply any kind of mask because it could affect negatively in different kind of facial expressions such as surprise that lower part of face move drastically in comparison with neutral state.

2.2 Feature Extraction

2.2.1 Local Binary Patterns (LBP)

Local Binary Pattern (LBP) is first proposed by Ojala et al. in 1996 [16]. It is an efficient texture operator which thresholds pixels in a neighborhood of a center pixel, and form a binary number to define the data. It has many advantages, perhaps the most important ones are its robustness to monotonic grey level changes from different lighting conditions and computational simplicity.

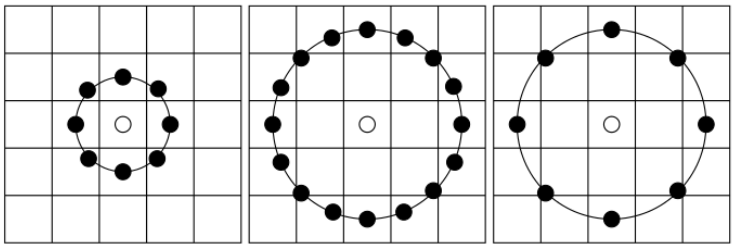

Figure 2.3: Basic LBP operator: circular (8,1), (16,2), and (8,2) neighborhoods.

Fig. 2.3 shows the basic LBP operator. Assuming small patches are extracted from an image I(x,y). In this patches, there are p neighbor pixels and a center pixel in this patches.

$$T = t(s(g_0 - g_c), s(g_1 - g_c), ..., s(g_{P-1} - g_c)) \tag{2.1}$$

$$s(x) = \begin{cases} 1 & , \quad x \geq 0 \\ 0 & , \quad x \leq 0 \end{cases} \tag{2.2}$$

Then, this binary value is converted into decimal, and it is stored LBP values of current block.

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{p-1} s(g_p - g_c) 2^p \tag{2.3}$$

There are several extension to this basic LBP operator. First, if the transition number in binary LBP code is 2 or less, this pattern is defined as *uniform pattern*. Ojala et al. showed that nearly %90 of natural textures in neighborhood of (8,1) and %70 for (16,2) are composed of uniform patterns. Similarly, in experiment using facial images,

%90.6 and %85.2 of (8,1) and (8,2) patterns respectively are uniform. Thus, uniform patterns are more discriminative and it could be a good alternative to define images using only uniform patterns. Some of the other extensions are rotation invariance and complementary contrast measure.

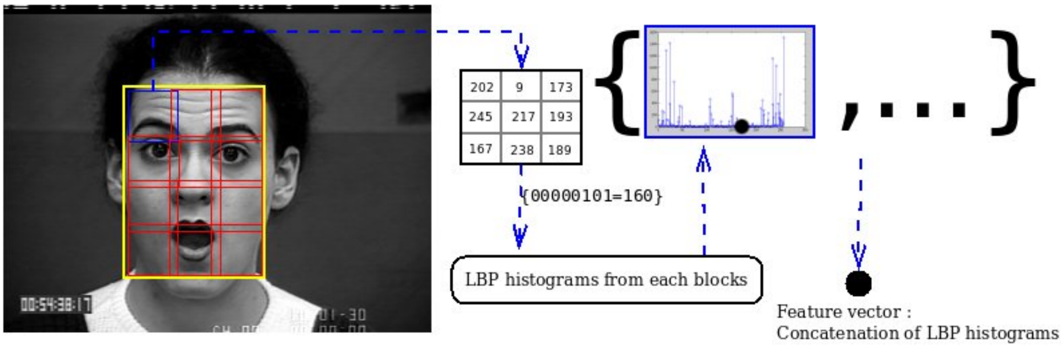

Figure 2.4: Example LBP feature extraction

Fig. 2.4 depicts how LBP features are extracted from images. After faces are detected and normalised, they are divided into small sub-partitions. These subregions could be overlapping or non overlapping. LBP is applied to each partition, and then image histograms are calculated from the results of LBP. For instance, each histogram in regions is in 256-element vector representation, when basic LBP is applied; whereas it is 59-element vector in uniform LBP. The histograms from subregions are concatenated into a single vector. This final repseruentation is the LBP features which is used in classification.

In last decade, LBP has gained a great popularity and there have been many proposed variants of LBP. Indeed, these variants are based on different preprocessing, neighborhood topology, thresholding, and multiscale analysis, or unification with other feature extraction methods. For variants of LBP and their reference papers, interested readers could consult to [25]. We prefered Local Ternary Patterns and Local Gabor Binary Patterns as alternative feature extraction methods to uniform LBP.

2.2.2 Local Ternary Patterns (LTP)

Local Ternary Patterns (LTP) has been proposed by Tan and Triggs [24], and it is proved to be more discriminative method that basic LBP operator, and highly robust to monotonic gray-level transformation. LTP bring threefold threshold, and it codes the image as a 3-valued function in specified neighborhood.

In LTP, gray-levels in a zone of width $\pm t$ around the center pixel i_c are quantized as zero, the ones above this threshold are $+1$, and the ones below are -1. LTP operator can be given as

$$LTP_{i,i_c,t} = \begin{cases} 1 & , \quad i \geq u_c + t \\ 0 & , \quad |i - i_c| \leq t \\ -1 & , \quad i \leq i_c - t \end{cases} \tag{2.4}$$

In LTP, threshold is defined by user. It is more resistance to noise, however loses its strict invariance to gray-level transformations.

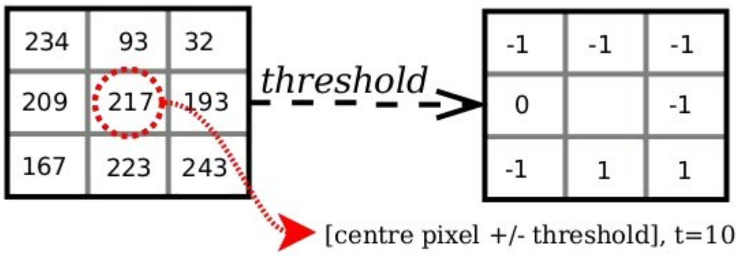

Figure 2.5: Extraction of local ternary pattern (LTP)

In Fig. 2.5, it can be seen how basic LTP operator is applied. LTP is a representation which takes image blocks and convert to a ternary pattern for each centre pixel. Here as example, the threshold is 10, and the pattern from 3 by 3 image window are calculated using Equation 2.4. These ternary patterns are splitted into two parts. The first is acquired taking 1's and setting the other pixels to 0. The second part takes -1's and sets the other pixels to 0. In this way, two different quantities are extracted from each centre point. Then, they are converted back to decimal.

The extraction of LTP as a feature vector is in same manner as applied in LBP. Images are divided into sub-partitions, and LTP is applied. Using two decimal values in each pixel, image histograms are calculated from subregions. It should be noted that two different image histograms will be extracted from each subregion. The concatenation of all histograms into a single vector will constitute the final feature vector representation of LTP.

2.2.3 Gabor Filters

Gabor features are one of the best performing features in texture analysis and face representation. It extracts the local informations from an image or region of interest and then combine them to form a descriptive feature. Even if it's mathematical roots go through a few decades ago, Daughman's paper [26] is accepted as a milestone, and this representation performs well even in today's computer vision and image processing applications.

Gabor filter can be used in feature extraction as 2D Gabor filter function:

$$\begin{aligned} \psi(x,y) &= \frac{f^2}{\pi\gamma\eta} e^{-\frac{f^2}{\gamma^2}x'^2 + \frac{f^2}{\gamma^2}x'^2} e^{j2\pi f x'} \\ x' &= x\cos\theta + y\sin\theta \\ y' &= -x\sin\theta + y\cos\theta \end{aligned} \tag{2.5}$$

This equation is the representation of Gabor filter in the spatial domain. Gabor is a complex plane wave which is multiplied by origin-centred Gaussian. f is the central frequency of the filter, Θ the rotation angle, γ sharpness along the Gaussian's major axis, η is sharpness along the minor axis. The aspect ratio of tha Gaussian is η/γ. The analytical form of previous equation in the frequency domain is

$$\begin{aligned} \Psi(u,v) &= e^{-\frac{\pi^2}{f^2}(\gamma^2(u'-f)^2 + \eta^2 v^{-2})} \\ u' &= u\cos\theta + v\sin\theta \\ v' &= -u\sin\theta + v\cos\theta \end{aligned} \tag{2.6}$$

Gabor filter is the simplified version of general 2D form derived from 1D elementary function by Daugman. This version enforces a set of similar filters which are scaled and rotated versions of each other regardless of frequency f and orientation θ. When it is used as a feature representation, Gabor features are generaly referred as Gabor banks, because it is used multiple application of filters in various frequencies and orientations.

In Gabor banks, the frequency corresponds to scale information and can be expressed as

$$f_m = k^{-m} f_{max}, \quad m = \{0, ..., M-1\} \tag{2.7}$$

where f_m is the m'th frequency, $f_0 = f_{max}$ is the highest frequency, and $k > 0$ is scaling factor. In same manner, the filter orientations are

$$\Theta_n = \frac{n2\pi}{N}, \quad n = \{0, ..., N-1\} \tag{2.8}$$

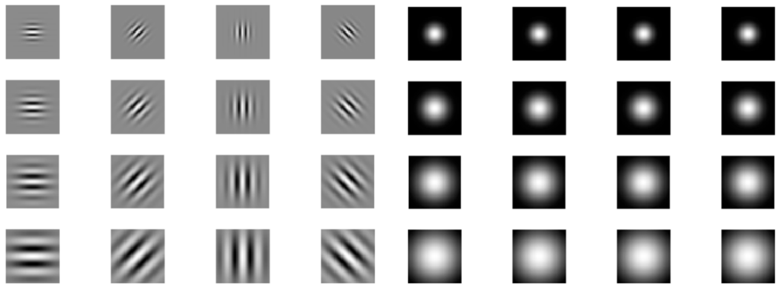

Figure 2.6: Gabor filters constructed in different scales and orientations. Real parts of gabor filter (left), magnitudes (right) (γ and η parameters are fixed: $\gamma = \eta = 1$)

In Figure 2.6, construction of Gabor filters in different orientations and scales is depicted. Real parts and magnitudes of Gabor filter is drawn in four different orientations and four scales. In implementation of Gabor filter as a feature vector, the number of scale and orientation can be changing into the specific task. We will restate our parameters in related experimental setup.

2.3 Classification

In this study, we essentially will use Support Vector Machines (SVM). Originally SVM was invented by Vladimir Vapnik, and current implementation with soft margin was proposed by Vapnik and Cortes in 1995 [27]. In this section, we will explain theoretical background of Support Vector Machines in brief.

2.3.1 Support Vector Machines

The support vector (SV) machine implements a mapping of input vectors to a high-dimensional nonlinear feature space in order to construct an optimal separating hyperplane. Support vector machines deal with two main issues: One is how we can find optimum separating hyperplane with a good generalization ability, other is the *curse of dimensionality*.

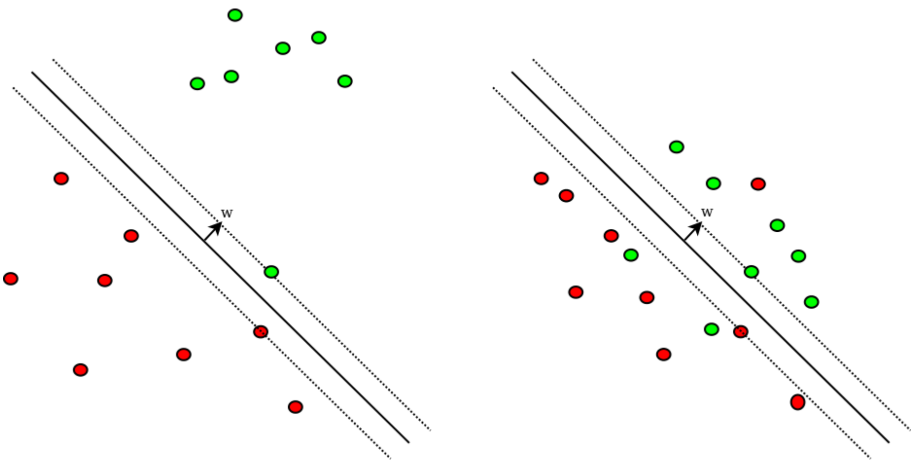

Figure 2.7: Classification : (a) poor generalization, (b) linearly inseparable case

Fig. 2.7 depicts an example of classification in poor generalization and linearly inseparable case. It can be clearly seem that generalization problem can be tackled by a $\Delta-$margin separating hyperplane and soft margin separating hyperplane. The Vapnik-Chervonenkis (VC) dimension of the set of Δ-margin separating hyperplanes with large Δ will be small. Thus, $\Delta-$margin separating hyperplane approach means better generalization. On the other hand, good generalization ability of maximal margin hyperplane is proved by Vapnik.

To understand support vector machines thoroughly, it will be proper to review the *optimal hyperplane* and Δ-*margin separating hyperplanes*.

2.3.1.1 The Optimal Hyperplane

For a given training data,

$$(X_1, y_1), ..., (X_l, y_l), \quad \in \mathbb{R}^n, \quad y \in \{+1, -1\}$$

a separating hyperplane can be defined as follows

$$(w \cdot x) - b = 0 \tag{2.9}$$

This approach tries to find *optimal hyperplane* and the distance from the nearest instances to the hyperplane is maximum. The same problem can be defined with a margin,

$$\begin{aligned} (w \cdot X_i) - b \leq 1 \quad if \quad y_i = 1, \\ (w \cdot X_i) - b \leq -1 \quad if \quad y_i = -1, \end{aligned} \tag{2.10}$$

We can rewrite the equalities as $y_i[(w \cdot X_i) - b] \geq 1$ and it is clear that the optimal hyperplane satisties this condition and minimizes

$$\Phi(w) = ||w||^2. \tag{2.11}$$

2.3.1.2 Δ-Margin Separating Hyperplanes

A hyperplane which satisfies the following conditions is called as a Δ-Margin Separating Hyperplanes:

$$\begin{gathered} (w^* \cdot X) - b = 0, \quad |w^*| = 1 \\ y = \begin{cases} 1, & (w^* \cdot X) - b \geq \Delta \\ -1, & (w^* \cdot X) - b \leq -\Delta \end{cases} \end{gathered} \tag{2.12}$$

Δ-Margin Separating Hyperplane with $\Delta = 1/|w^*|$ is the optimal hyperplane. Following theorem is true and it is the answer for why generalization ability of Δ-margin separating hyperplane which is used in SVM is better.

$x \in X$ belong to a sphere of radius R. Then, a set of Δ-margin separating hyperplanes has VC dimension h bounded the inequality

$$h \leq \min\left(\left[\frac{R^2}{\Delta^2}, n\right]\right) + 1 \tag{2.13}$$

The VC dimension of a separating hyperplane seems n+1, where n is the dimension of the space. However, support vector machines with Δ-margin approach can separate with relatively smaller estimate of VC dimension.

To summarize, we construct the decision functions transforming the input space into the convolution of the inner products,

$$f(x) = sign\left(\sum_{support\ vectors} y_i \alpha_i K(x_i, x) - b\right) \tag{2.14}$$

Thinking about seperable case, the maximum of the functional can be written as

$$W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{2.15}$$

subject to the constraints

$$\sum_{i=1}^{l} \alpha_i y_i = 0, \qquad \alpha_i \geq 0, \qquad i = 1, 2, ..., l. \tag{2.16}$$

Looking into the functional in Eq. 2.15, it is written the convolution of inner products $K(x_i, x)$ instead of inner products x_i, x. It is named as *kernel fuction* and the application of this function could makes perfect classification especially in inseparable case. Typical examples of kernel functions are homogeneous or inhomogeneous polynomial functions, radial basis functions (RBF), hyperbolic tangent, and two layer neural networks.

$$
\begin{aligned}
\textbf{Polynomial:} \quad & K(x,x_i) = [(x \cdot x_i)]^d. \\
\textbf{Polynomial:} \quad & K(x,x_i) = [(x \cdot x_i) + 1]^d. \\
\textbf{Radial basis:} \quad & K_\gamma(|x - x_i|) = exp(-\gamma|x - x_i|^2) \\
\textbf{Hyperbolic tangent:} \quad & K(x_i,x_j) = |\tanh(kx_i \cdot x + c) \\
& \textit{for some } k > 0 \textit{ and } c < 0
\end{aligned}
\qquad \textbf{(2.17)}
$$

If it is necessary to apply a kernel function, depends on the nature of the feature vectors. Thus, trying different kernel functions sweeping parameters could help to find best performance SVM classifier.

3. EVALUATION

In this chapter, evaluation systems and the facial expression databases which are used in this study will be explained in detail. Most datasets are publish with well-defined databases. We will explain their experimental setup and evaluate our methods using these protocols.

3.1 Types of classification

In machine learning, classification task could be divided into four categories: *binary, multi-class, multi-label* and *hierarchical*.

In binary classification, inputs are classified into only one class of two non-overlapping classes. It is the most used classification method, and other classification methods can be transformed into binary problem. For instance, gender classification in face analysis is an example of binary classification.

Multi-class classification is a mapping of inputs into only one class and classes are non-overlapping. Indeed, multi-class classification could be done using either multi-class classifiers or binary classification as one-versus-all(OvA) or all-versus-all(AvA). The performance of OvA and AvA could change according to the dimension of the feature space and the number of inputs. Facial expression classification is a multi-class task, and we will use OvA approach in our evaluation.

In multi-class classification, the input is to be classified into several non-overlapping classes. Each instances could be assigned into multiple labels. On the other hand, there are various superclasses and subclasses in hierarchical approach. This hierarchy is defined and not changed during classification For example, there are many hierarchical tasks in text classification and bioinformatics.

3.2 Performance Metrics

3.2.1 Binary case

For a two-class (binary) classification task, performance metrics could be written using the confusion matrix in Table 3.1. In following equations, true positive, false negative, false positive, and true negative are stated with their abbreviations, TP, FN, FP, and TN, respectively.

Table 3.1: Confusion matrix for binary classification

		Predictions		
		A	*B*	*total*
Actual Class	*A*	True Positive	False Negative	Pos
	B	False Positive	True Negative	Neg
	total	P	N	

The overall effectiveness of a classifier could be understood by *accuracy* which is a fraction of correctly classified instances.

$$accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{3.1}$$

Higher accuracy is an indication of increased classification performance. However, there are other important measures such as *precision* and *recall*. Precision is class agreement of class labels with positives given by the classifier. On the other hand, recall, in other word sensitivity is the effectiveness of a classifier to identify positive instances.

$$\begin{aligned} precision &= \frac{TP}{TP+FP} \\ recall &= \frac{TP}{TP+FN} \end{aligned} \tag{3.2}$$

For a powerful evaluation, precision and recall should be considered in combination. Another performance metric is F score which makes a connection between the data's positive labels and those given by the classifier neglecting the true negative predictions.

$$F\ score = \frac{(\beta^2+1)TP}{(\beta^2+1)TP+\beta^2 FN+FP} \tag{3.3}$$

The traditional or balanced F-score is the harmonic mean of precision and recall. Thus, it is named as F_1 score and β is 1.

One of the most common performance metrics is the *receiver operating characteristic* (ROC) curve. It is graphical representation by true positive rate (recall) and false positive rate. False positive rate can be defined as

$$False\ positive\ rate = \frac{FP}{FP+TN} \tag{3.4}$$

In ROC curve, *y=x* diagonal equals to random prediction, whereas leaning towards (0,1) point is perfect classification. In any steps, it can be said that the curve is worse than random classification if the line is under the *y=x* diagonal.

3.2.2 Multi-class case

Facial expression classification is a multi-class problem. So, we cannot calculate the performance metrics as in previous subsection.

Table 3.2: Confusion matrix for multi-class classification

	1	2	3	4	5	6	7
1	n_{11}	n_{12}	n_{13}	n_{14}	n_{15}	n_{16}	n_{17}
2	n_{21}	n_{22}	.	.	.	.	.
3	n_{31}	.	.				
4	n_{41}	.		.			
5	n_{51}	.			.		
6	n_{61}	.				.	
7	n_{71}	.					.

Confusion matrix for multi-class classification is depicted in Fig. 3.2. It is similar to our task in facial expression analysis. Vertical axis is ground truth, and horizontal axis is predictions. Looking into class 1, it can be clearly seen that true positive is n_{11}, the sum of $n_{21}, n_{31}, n_{41}, n_{51}, n_{61}$ and n_{71} is false positive, the sum of $n_{12}, n_{13}, n_{14}, n_{15}, n_{16}$ and n_{17} is false negative.

We can write true positive (TP), false negative (FN), false positive (FP), and true negative (TN) in generic form as follows,

$$TP(i) = n(ii) \tag{3.5}$$

$$FN(i) = \sum_{i \neq j} n_{ij} \tag{3.6}$$

$$FP(i) = \sum_{i \neq k} n_{ki} \tag{3.7}$$

$$TN(i) = \sum_{j \neq i, k \neq i} n_{jk} \tag{3.8}$$

Using these definitions, we can write the performance metrics for each classes,

$$accuracy(i) = \frac{TP(i) + TN(i)}{TP(i) + FP(i) + TN(i) + FN(i)}, \tag{3.9}$$

$$precision(i) = \frac{TP(i)}{TP(i) + FP(i)}, \tag{3.10}$$

$$recall(i) = \frac{TP(i)}{TP(i) + FN(i)}, \quad and \tag{3.11}$$

$$F_{score}(i) = \frac{(\beta^2 + 1)TP(i)}{(\beta^2 + 1)TP(i) + \beta^2 FN(i) + FP(i)} \tag{3.12}$$

3.3 Datasets

In this section, facial expression datasets which is used in this study will be explained. We evaluated the performance of our methods using four datasets: the *Extended Cohn-Kanade Dataset*, *MMI Facial Expression Database, JAFFE database,* and *Static Facial Expressions in the Wild (SFEW) database*.

3.3.1 The Extended Cohn-Kanade (CK+) Dataset

The extended Cohn-Kanade (CK+) dataset is improved version of the Cohn-Kanade database which was released in 2000, and have been used in automatic detection of individual facial expressions. In CK+, the number of sequences are increased by 22% and the number of subjects by 27%.

This dataset is consist of face images of 210 adults. The participants vary 18 to 50 years of age, 69% female and different ethnicities (81% Euro-American, 13% Afro-American, and 6% other groups). Participants are instructed to imitate single action unit and combinations of action units.

Figure 3.1: Examples from CK+ database [1]
Emotion-labeled images: (a) Disgust, (b) Happy, (c) Surpise, (d) Fear, (e) Angry, (f) Contempt, (g) Sadness, (h) Neutral.

The dataset is made up of action unit (AU) and discrete emotion labels. The emotion classes are angry, contempt, disgust, fear, happy, sadness, and surprise. The total number of emotion labeled images from these classes are 327. There are neutral class, however we will not use nuetral images, because it is not used in emotion detection results of [1]

Table 3.3: Number of emotion-labeled images in CK+

Emotion	**N**
Angry (An)	45
Contempt (Co)	18
Disgust (Di)	59
Fear (Fe)	25
Happy (Ha)	69
Sadness (Sa)	28
Surprise (Su)	83

In Figure 3.1, it can be seen images from different emotion classes in CK+ database. If a sequence is emotion-labeled, the emotion-labeled ones are generally the first neutral image and the last one which represents the apex point of each facial expression. The number of samples from each emotion class is given in Table 3.3.

3.3.2 The MMI Facial Expression Database

The MMI Facial Expression Database [5, 28] has been developed by Maja Pantic, Michel Valstar and Ionnias Patras in 2002. It contains the full temporal activity of human face from neutral to a specific emotional state and back to the neutral state. MMI database have annotated with both FACS and discrete emotions, and there are video and static images from different sources in varying resolutions and image size. The database consists of over 2900 videos and high resolution still images of 75 person. It is accessable via internet to the scientific community, and a number of images from the database can be retriaved using a variety of physical attributes.

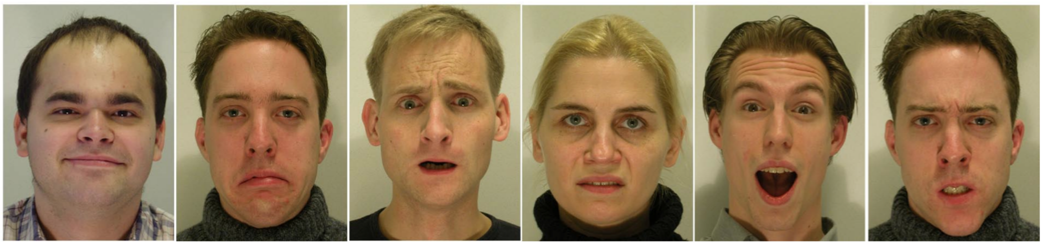

Figure 3.2: Examples of static frontal face images from the MMI Facial Expression Database

In Figure 3.2, sample still images from the MMI database can be seen. In video sequences of the database, the emotional state of faces are supported. The neutral state in the beginning is named as *onset*. Then, the emotion starts to appear and the moment at emotions is the most intensive is *apex* point. Through the end of sequence, the face returns to its neutral state which is named as *offset* in this setup. As these are standardised, it is possible to use instances from various intensities of facial expressions for the interested researchers.

As well as frontal images and videos, the database consists of profile-face or mirrored profile sequences at the same time. Figure 3.3 shows the profile-face images and automated facial fiducial points tracking in these sequnces of these sequences from the MMI database.

The samples from different subjects are acquired in this way: The subjects are asked to send a neutral expression and imitate a specific AU combination or emotion. There are minimal out-of-plane head movements and the subjects are instructed by

Figure 3.3: Automated facial fiducial points tracking in profile-face image sequences contained in the MMI Facial Expression Database [5]

an experienced FACS coder. After acquiring the whole data, the selection of instances are made by two FACS coder. Thus, the ones which is classified same by two experts are added to the database.

In comparison with the Extended Cohn-Kanade database, the MMI database has more diversive resources and annotations, however it is another laboratory controlled database. So, it is good to evaluate our algorithms, but does not reflect the behaviour of our system in real-world environment.

A disadvantage of the MMI database, there is no officialy proposed benchmark and a baseline as CK+. So, it is unfortunately not possible to compare our results with previously reported publications. It might be because of online structure of database, and addition of new instances from the initial release date. Thus, we searched for emotion-labeled image sequences and took examples taking into consideration onset, apex, and offset of facial expressions, and created our own experimental setup.

3.3.3 The Japanese Female Facial Expression (JAFFE) Database

The Japanese Female Expression (JAFFE) Database contains 213 images of ten posers, 3 or 4 examples of each of the seven basic expressions (happiness, sadness, surprise, anger, disgust, fear, and neutral). Sample images from the database are depicted in Fig. 3.4 below.

Figure 3.4: Samples from the Japanese Female Facial Expression (JAFFE) Database

Each image has a resolution of 256 x 256 pixels, and grayscale. Even if it is one of the pritimitive datasets in facial expression recognition, we prefered to use in order to compare our results with previous works which have used the JAFFE protocol.

3.3.4 The Static Facial Expressions in the Wild (SFEW)

In both other face analysis tasks and facial expression recognition, the usage of realistic data is fairly important in building more robust system to various challenges such as occlusion, illimunation change or pose variations. For this reason, we used another dataset which is composed of shots from movies. While movies are generally shot in somewhat controlled enviroments, studios, they at least provide nearly world environment in comparison with images recorded in lab environments.

The Static Facial Expressions in the Wild (SFEW) database [29] was developed by selecting frames from AFEW database (the one which is composed of videos). These frames contains unconstraint facial expressions, varied head poses, large age and ethnicity range, and nearly real world illumination conditions. There are 700 images and these are labeled for six basic expressions *angry, disgust, fear, happy, sad, surprise,* and the *neutral* by two independent labellers.

Figure 3.5: Samples from the Static Facial Expressions in the Wild (SFEW) Database

In Fig. 3.5, sample images from SFEW database are given. In AFEW protocol, there are three categories: *strictly person specific (SPS), partial person independent (PPI),* and *strictly person independent (SPI)*. Standard SFEW protocol falls under SPI category.

3.4 Experimental Setup

In this section, we will present the experimental setup for our evaluation. To make a comparison with previous works which uses the same datasets, it will be proper to use same experimental setup as applied in these studies. So, we used the benchmarking protocol and evaluation metric in CK+ reference paper [1]. In similar manner, we set similar experimental settings for the Web Images Database [30]. To summarize, we can present our experimental setup and evaluation criteria as follows,

- In the Extended Cohn-Kanade (CK+) Dataset, we used 327 emotion-labeled images. These 327 images are of 118 person. It is proposed to conduct person-independent training and testing in standard benchmark. That is to say, all images from a person is leaved out, and other images are trained. Leaved-out images are used as test procedure. This process is repeated 118 times. This methos maximize the test and training amount of data, and named as "one-subject-out cross validation" in the literature. At the end, a confusion matrix which contains the results of all classes is created.

- In the MMI Facial Expression Database, we filtered image sequences which have emotion annotations, and found 236 video files. Using these data, we extracted the frames in the middle of each sequence, with the knowledge of that the apex points are nearly in the middle. There are 1304 images which are used in our experiments in the MMI database. This images are from six classes: anger, disgust, fear, joy, sadness, and surprise. We mixed this selection, and used 70% of training purposes and 30% of each class for testing and performance evaluation.

- In JAFFE database, there is no standardised protocol for evaluation and performance measurement. Thus, we will reconsider it in following section.

- In the SFEW database, the strictly person independent (SPI) protocol is applied. The database is divided into two sets. Each sets has seven subcategories corresponding to the seven expression classes. There are 346 images in Set 1, and 354 in Set 2. The total number of subjects is 95 in the database. The protocol should be as follows: first, train on set 1 and test on set 2 and then train on set 2 and test on set 1. The evaluation metric in this database is different from the previous ones. The performance of FER systems are *accuracy, precision, recall,* and *specificity.*

3.5 Results

In this section, we will compare our results with previous studies which used the Extended Cohn-Kanade database and the Web Image database. We will start with the Extended Cohn-Kanade (CK+) database evaluation.

3.5.1 The Experimental Results in CK+ Database

Looking into the reference paper of CK+ database [1], we will review the benchmark results which is presented. In discrete emotion detection, the performance of the shape (SPTS) and the canonical appearance (CAPP) features are evaluated. In classification, the linear kernel SVM is used. It can be supposed that complicated kernel functions perform better in all cases. However, it is not necessary to use in case of lower number of instances and feature dimension. Besides, the generalization ability of linear kernel is comparably good in unseen data. Because the best results are achieved using the fusion of these two features, SPTS and CAPP, we depict these results in Table 3.4.

Table 3.4: Confusion matrix for the SPTS+CAPP features and linear SVM [1]

	Anger	Disgust	Fear	Happy	Sadness	Surprise	Contempt
Anger	**75.0**	7.5	5.0	0.0	5.0	2.5	5.0
Disgust	5.3	**94.7**	0.0	0.0	0.0	0.0	0.0
Fear	4.4	0.0	**65.2**	8.7	0.0	13.0	8.7
Happy	0.0	0.0	0.0	**100.**	0.0	0.0	0.0
Sadness	12.0	4.0	4.0	0.0	**68.0**	4.0	8.0
Surprise	0.0	0.0	0.0	0.0	0.0	**96.0**	0.0
Contempt	3.1	3.1	0.0	6.3	3.1	0.0	**84.4**

According to the results in Table 3.4, three classes; happy, surprise, and disgust are the ones which have fairly high accuracies in comparison with remaining four classes. Besides, sadness can be classified as anger mistakenly. The worst performance is observed in fear and sadness classes.

Moving onto the similar performance evaluations on CK+ database using the feature extraction methods that we used. In [2], a hierarchical approach is adopted in the classification of person-independent facial expressions. This study concentrates on two tier classification scheme. In first scheme, the easily-confused prototypic expressions are considered as one class, and the remaining are summed into another group. In the second tier, the merged class in the first scheme are trained and classified. In this study, the selected easily-confused classes in the first tier are anger and sadness.

In feature extraction process, local binary pattern (LBP) and displacement features are used. In two-tier scheme, these two features are combined. Because we used one-tier scheme and LBP features in our study, we take the LBP results of this study as another benchmark for evaluating our results. In Table 3.5, the results of LBP is given. Even if the same procedure which is defined in CK+ database's benchmark is applied, the class of contempt is not included in [2].

Table 3.5: Confusion matrix for the LBP feature [2]

	Anger	Disgust	Fear	Sadness	Happy	Surprise
Anger	**46.2**	2.5	0.4	47.5	1.5	1.9
Disgust	1.6	**86.1**	2.4	9.1	0.0	0.8
Fear	0.0	3.7	**85.6**	10.2	0.0	0.5
Sadness	1.5	1.5	1.5	**94.4**	0.0	1.0
Happy	0.8	1.3	6.1	1.3	**89.2**	1.3
Surprise	0.2	1.2	1.6	3.5	0.2	**93.3**

In our experiments, we applied local binary pattern (LBP) and local ternary pattern (LTP) features with linear SVM classifier in CK+ evaluation protocol which is explained in previous section. Our confusion matrices for LBP and LTP are displayed below in Tables 3.6 and 3.7.

Table 3.6: Confusion matrix for LBP $(8,2)_{ue}$ + Linear kernel SVM

	Anger	Disgust	Fear	Happy	Sadness	Surprise	Contempt
Anger	**93.33**	2.22	0.00	0.00	4.44	0.00	0.00
Disgust	1.69	**98.31**	0.00	0.00	0.00	0.00	0.00
Fear	4.00	0.00	**76.00**	8.00	0.00	8.00	4.00
Happy	0.00	0.00	0.00	**100.**	0.00	0.00	0.00
Sadness	25.00	3.57	0.00	0.00	**67.86**	3.57	0.00
Surprise	0.00	0.00	1.20	0.00	0.00	**97.59**	1.20
Contempt	0.00	0.00	5.56	0.00	5.56	0.00	**88.89**
ACC	96.33	98.81	95.29	98.73	93.65	97.80	97.44

Table 3.7: Confusion matrix for LTP $(8,2)_{ue}$ + Linear kernel SVM

	Anger	Disgust	Fear	Happy	Sadness	Surprise	Contempt
Anger	**97.78**	0.00	0.00	0.00	2.22	0.00	0.00
Disgust	1.69	**98.31**	0.00	0.00	0.00	0.00	0.00
Fear	0.00	0.00	**80.00**	8.00	0.00	8.00	4.00
Happy	0.00	0.00	0.00	**100.**	0.00	0.00	0.00
Sadness	25.00	3.57	0.00	0.00	**67.86**	3.57	0.00
Surprise	0.00	0.00	1.20	0.00	0.00	**97.59**	1.20
Contempt	5.56	0.00	5.56	0.00	5.56	0.00	**83.33**
ACC	94.77	99.16	95.89	98.74	93.99	97.81	96.62

The performance of our 7-class emotion classification using LBP and LTP feature vectors are depicted as confusion matrices in Tables 3.6 and 3.7. It can be seen that the recognition rates of LBP features are higher than previous works that we reviewed. The easily mis-classified classes are fear and sadness. According to our results, recognition rates of fear and sadness are 88% and 82.14% respectively in LBP features.

Moving onto the CK+ database evaluation using Gabor filter in the extraction of features. After registration and normalisation of images, faces are scaled into 128 by 128 pixel. Then, Gabor filters are applied in 5 scales and 8 orientations. Similary, SVM is used in classification.

Table 3.8: Confusion matrix for Gabor filter + Linear kernel SVM

	Anger	Disgust	Fear	Happy	Sadness	Surprise	Contempt
Anger	**93.33**	0.00	2.22	0.00	4.44	0.00	0.00
Disgust	5.08	**94.92**	0.00	0.00	0.00	0.00	0.00
Fear	8.00	4.00	**76.00**	4.00	4.00	4.00	0.00
Happy	0.00	0.00	0.00	**100.**	0.00	0.00	0.00
Sadness	10.71	0.00	0.00	0.00	**89.29**	0.00	0.00
Surprise	0.00	0.00	0.00	0.00	0.00	**98.80**	1.20
Contempt	5.56	0.00	0.00	0.00	1.11	5.56	**77.78**
ACC	94.58	98.58	96.00	99.14	95.41	98.32	96.19

In conclusion, we reached these results in our performance evaluation in CK+ database:

1. Our local binary pattern (LBP)-based facial classification results are better than the results which was reported in previous works using same feature extraction method. In the performance of local binary pattern features in classification, its robustness to environmental changes such as lighting is a very good advantage in application to face analysis. Besides, it conserves the locality information, if it is used in proper neighborhood and blocking the images into partitions. According to our results, it is possible to reach higher performance without increasing the feature length unnecessarily. Partitioning image in an optimal way is important, because our application should be applicable for analysis of image sequences.

2. Local Ternary Pattern (LTP) is one of the LBP variants which changes the thresholding approach. In [24], it is reported that LTP performs better in face recognition. We used LTP using same neighborhood and blocking strategy and compared its performance with LBP results. Although it is expected to increase the recognition rates of seven-class facial expression classification, our LBP results are better than LTP. Denifing the right threshold is of utmost importance in the performance of LTP. However, our assumption failed because our database composed of very limited number of images, and LBP is necessary to expresss lighting change of these images.

3. The number of emotion-labeled images in the Extended Cohn-Kanade database is unbalanced. The number of images per class changes from 18 to 69. Soma classes have very limited samples, ans this decreases the classification performance of these classes.

3.5.2 The Experimental Results in the MMI Database

In MMI Database, 1304 images are used and we applied creating learning model and testing 10 times in each time randomly mixing the image sets and using 70% training, 30% test purposes.

Table 3.9: Confusion matrix for LBP $(8,2)_{ue}$ + Linear kernel SVM in the MMI database

	Anger	Disgust	Fear	Happy	Sadness	Surprise
Anger	**62.40**	0.40	0.40	0.40	1.40	0.00
Disgust	1.80	**26.00**	0.00	0.70	0.30	0.20
Fear	0.60	0.80	**60.20**	0.10	0.10	2.20
Joy	0.10	0.40	0.40	**77.70.**	0.80	0.60
Sadness	13.0	0.40	0.10	0.30	**73.70**	0.20
Surprise	0.00	0.30	1.20	0.50	0.00	**82.00**

Table 3.9 shows the averaged number of predictions in 10 experiments which is mixed training and test groups randomly.

3.5.3 The Experimental Results in JAFFE Database

In JAFFE database, there is not any standardised benchmark or protocol. Thus, we conducted our experiments in this way : In database, there are 3 or 4 images of each posers in seven basic emotions. We picked the poser's first two images of each emotions to use for training purposes. The remaining of the images are in test set.

From 213 images, the SVM models are trained using the extracted features from 140 images, and then the remaining 73 images are for reporting the performance.

Table 3.10: Confusion matrix for LBP $(8,1)_{ue}$ + Polynomial kernel SVM in JAFFE database

	Anger	Disgust	Fear	Happy	Neutral	Sadness	Surprise
Anger	**9**	1	0	0	0	0	0
Disgust	0	**8**	1	0	0	0	0
Fear	0	1	**11**	0	0	0	0
Happy	0	0	0	**11**	0	0	0
Neutral	0	0	0	0	**10**	0	0
Sadness	0	0	1	1	0	**9**	0
Surprise	0	0	0	0	0	0	**10**

In Table 3.11 first the face region is cropped using the position of eyes in each images, and then histogram equalization is applied to these faces. Local binary patterns (LBP) are extracted within 3 by 4 subregions and concatenated into a vector as feature to define the related facial expression. We used multiclass support vector machines (SVM) with different kernel functions. Whereas the classification performance is fairly poor in radial basis function (RBF) and sigmoid kernels, the 7-class classification accuracy is 91.78% and %93.15 for linear and polynomial kernel functions respectively. For this reason, we used polynomial kernel multiclass SVM in our results in Table 3.11.

3.5.4 The Experimental Results in the SFEW Database

The Static Facial Expressions in the Wild (SFEW) database uses strictly person independent protocol, and from the two sets in database, both of them are used in training and testing. The evaluation metrics will be different from the other datasets' results. In reference paper of SFEW [29], the performance of the systems are defined as *accuracy, precision, recall,* and *specificity.*

Before our results on SFEW database, let's review the definition of these metrics which are defined as a part of SPI BEFIT challenge in [29] briefly.

$$
\begin{aligned}
OverallAccuracy &= \frac{tp+tn}{tp+fp+fn+tn} \\
ClassWisePrecision &= \frac{tp}{tp+fp} \\
ClassWiseRecall &= \frac{tp}{tp+fn} \\
ClassWiseSpecificity &= \frac{tn}{tn+fp}
\end{aligned}
\tag{3.13}
$$

Here, tp, fp, tn, and fn are true positive, false positive, true negative, and false negative respectively.

Table 3.11: Average expression classwise Precision, Recall and Specifity results on the SFEW database based on the SPI protocol

	Anger	Disgust	Fear	Happy	Neutral	Sadness	Surprise
Precision	0.42	0.09	0.20	0.56	0.33	0.15	0.21
Recall	0.32	0.19	0.18	0.52	0.22	0.21	0.35
Specificity	0.70	0.72	0.70	0.74	0.72	0.69	0.72

Extracting LBP features from histogram equalized images and applying SPI protocol with linear kernel mutliclass SVM, seven class accuracy for SFEW is 59.76%. Appearantly, it is pretty lower than the results of similar experiments on CK+, MMI, and JAFFE databases. The reason of this difference is due to the close to the real world conditions in the SFEW database.

Using the same feature extraction method, the effect of normalization in overall accuary is very limited in previous datasets. However, it causes approximately +/- 4% chance in SFEW experiments. Whereas it is nearly unobservant in previous lab controlled datasets, SFEW database is more preferable to simulate the real conditions and test the performance of our classifier in uncontrolled environment.

3.6 Comments

In this section, the types of classification and performance metrics are covered at first. Then, the used datasets are briefly described and the setups of our experiments are given in detail.

We conducted several experiments on the Extended Cohn-Kanade(CK+), MMI Facial Expression, JAFFE, and SFEW databases. In the CK+ database, all feature extraction methods which is covered in previous section are applied. These results are done in computer, and we reached to some conclusions in order to consider in our embedded implementation of facial expression recognition system:

- Most of the standardised benchmarks and datasets reached to saturation. That is to say, they are useless to understand and evaluate the performance of different classifiers. Normally, there should be more drastic difference among the results of LBP, LTP, and Gabor filter in CK+ database, but we can see that it is necessary to find a firmer evaluation protocol instead of using these datasets.

- Besides LBP and LTP, we reported the results using Gabor filter, however it's performance is not too much better than the other methods, even if the feature length in Gabor filter is considerably longer than LBP and its variants. On the other hand, LBP is computationally easier than applying Gabor filters.

- Histogram equalization and Tan & Triggs normalization (gamma correction and DoG) are used and both of them are efficient to increase the classification performance. For instance, if we used the pixels of images with PCA dimension reduction, it would be more invariant to lighting conditions and the normalization or preprocessing might be more significant. But, it can be said that LBP and variants are roburst to lighting changes. Nonetheless, our experiments showed that the normalization, more or less, has a positive effect on the total performance of the facial expression recognition system.

4. IMPLEMENTATION

In this chapter, the functional characteristics of our development platform will be described. We made a facial expression classification system using a few alternative methods. After the performance evaluation, proper methods are selected via embedded approach. Our aim is to achieve our goal to make an embedded system which is capable of analyzing and classifying facial expressions of humans.

4.1 Target Platform

Our target platform is the *Zedboard*. It is an evaluation and development board based on the Xilinx's Zynq-7000 All Programmable System-on-Chip (SoC). The Zedboard has two Cortex-A9 ARM processor and 85,000 Series-7 Programmable Logic (PL) cells. In Fig. 4.1, the general placement of Zedboard can be seen.

Having both power efficient, high performance ARM architecture and Field Programmable Gate Arrays (FPGAs) on same board make it an ideal platform for hardware and software developers. Heterogeneous computation introduces an advantage to choose hardware accelerated operations in some cases.

Image and video processing applications are used everywhere, especially automotive industry, security, and surveillance. However, the complexity of the tasks can not catch up the features of target hardware because these platforms are generally lack of energy and computational resources. Our task is to run our facial expressions recognition system using our pre-trained models as a user-end application.

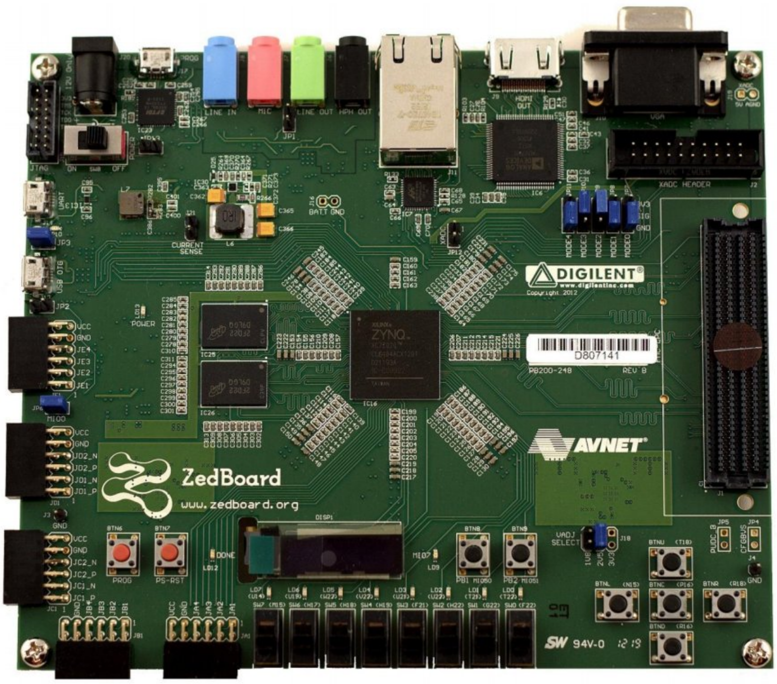

Figure 4.1: *Zynq^{TM}* Evaluation and Development Board

The peripheral and extension capabilities which is provided by Zedboard consist of:

- **Xilinx ® XC7Z020-1CSG484CES EPP**

  Primary configuration = QSPI Flash

  Auxiliary configuration options

  Cascaded JTAG

  SD Card

- **Memory**

  512 MB DDR3 (128M x 32)

  256 Mb QSPI Flash

- **Interfaces**

  USB-JTAG Programming using Digilent SMT1-equivalent circuit

  Accesses PL JTAG

  PS JTAG pins connected through PS Pmod

  10/100/1G Ethernet

  USB OTG 2.0

SD Card

USB 2.0 FS USB-UART bridge

Five Digilent Pmod™ compatible headers (2x6) (1 PS, 4 PL)

One LPC FMC

One AMS Header

Two Reset Buttons (1 PS, 1 PL)

Seven Push Buttons (2 PS, 5 PL)

Eight dip/slide switches (PL)

Nine User LEDs (1 PS, 8 PL)

DONE LED (PL)

- **On-board Oscillators**

  33.333 MHz (PS)

  100 MHz (PL)

- **Display/Audio**

  HDMI Output

  VGA (12-bit Color)

  128x32 OLED Display

  Audio Line-in, Line-out, headphone, microphone

- **Power**

  On/Off Switch

  12V @ 5A AC/DC regulator

- **Software**

  ISE® WebPACK Design Software

  License voucher for ChipScope™ Pro locked to XC7Z020

4.2 Linux on Embedded Platforms

Embedded systems are used in many applications such as consumer electronics, networking, machine control, automation, navigation equipment, aerospace, and biomedical. Most of these systems are based on an operating system installed on a linux kernel.

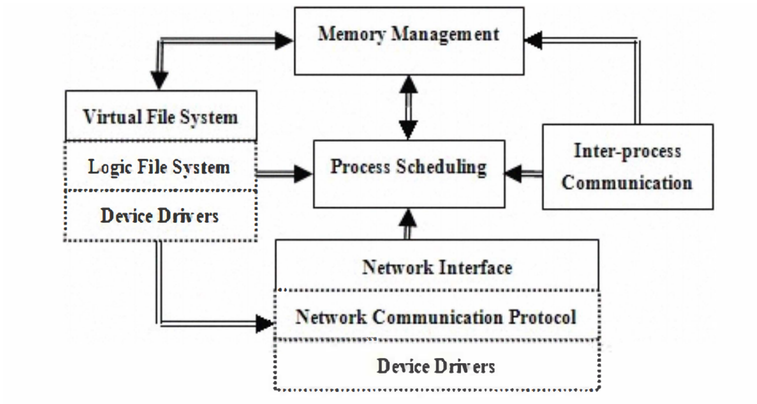

Figure 4.2: Linux kernel modules and relationship diagrams [6]

Figure 4.2 shows the general structure of a linux kernel. The kernel consists of five main subsystems. These are process scheduling module, memory management module, the file system module, inter-process communication module, and the network interface module. To cover the linux kernel and related key technologies for embedded linux on ARM, interested readers could consult to [6].

Embedded linux operating systems can be used compiling anyone's own kernel on a specific hardware. Indeed, it could be good to write our own kernel for a specific application or device. However, it will be more convenient way to use the kernel which is given in our embedded board's producers in order to ensure a working operation system to use as a prototype of an embedded vision implementation. We preferred to install our operating system using pre-compiled kernel by Digilent. [1]

In installation steps, the guide papers which are published by Digitent, Avnet, and Analog Devices are used, and the detailed steps how to format and partition the SD

[1]http://www.digilentinc.com/Data/Products/EMBEDDED-LINUX/ZedBoard_GSwEL_Guide.pdf

card, and install the operating system, device tree, and kernel files are described in Appendix-A.

Before a literature survey on embedded application of facial expression recognition (FER) systems, and our implementation, it would be good to address the definition of *real-time* for the benefit of anyone who is unclear about its meaning, and comment if our application is real-time or not.

Real-time means that a given stimulus must elicit an appropriate response from the system and/or application quickly enough to be satisfactory. From this perspective, there are two categories of real-time requirement: One is **hard real-time** requirement which is defined as a scenario in which the failure of application or system to respond to a given external stimulus within the specific interval will result in a catastrophic failure of the system or application. The second is a **soft real-time** requirement which is defined as a scenario in which the failure of the system and/or application to respond to a given external stimulus within the specified interval will result in a recoverable error or degredation in the performance, but not result in a catastrophic failure.

An application or a sytem can be evaluated according to their latency and determinism. Moving onto the definition of these concepts:

- **Latency** is the observed, measured responsiveness of the system and/or application to external stimuli. That is to say, how the system and/or application send an appropriate response to a given stimulus can be measured by latency.

- **Determinism** is the predictability of the system and/or application to respond to an external stimuli within a specified interval of time. It is the measure of a system or application how can consistently produce response within the specified interval.

  To evlauate the performance of a real-time system, both latency and determinism must be measured rather than calculation. It can be said that latency is the common concept in both hard and soft real-time requirement. However, determinism is much more important in hard real-time requirements. There are various ways to reduce latency and improve the determinism in embedded linux systems, however we put a stop to discussion of real-time because our implementation is based on precompiled kernels.

4.3 Previous Works and Literature Survey

In this section, we will concentrate on previous works on the embedded implementations on facial expression or emotion classification problem. Before reporting the performance measures and evaluation of our implementation, a few previous design which is related to our main scope, *emotion analysis* will be considered. Most target platforms are different and a fair comparison among these implementations and ours seems impossible. However, it will be preferable to evaluate all these designs in terms of preferred target platforms, timing performance, applied tasks, and so on.

To the best of our knowledge, most noteworthy embedded examples of facial expression classification system are given below, and these works are reconsidered briefly, because of their importance and relevance to our implementation:

1. Yang-Bok Lee, Seung-Bin Moon, and Yong-Guk Kim *Face and Facial Expression Recognition with an Embedded System for Human-Robot Interaction*, [7]

2. Paul Santi-Jones, Dongbing Gu, *Static Face Detection and Emotion Recognition with FPGA Support* , [8]

3. Te-Feng Su et al., *Automatic Facial Expression Exaggeration System with Parallelized Implementation on a Multi-Core Embedded Computing Platform*, [9]

4. Neng-Sheng Pai, Shih-Ping Chang, *An Embedded System for Real-time Facial Expression Recognition based on the Extension Theory*, [31]

In literature, we could not find a similar implementation which is done using our target platform. Looking into these previous works, they varies in terms of scale and abilities of target platforms, power consumptions, time performance, implementation details and tasks, but within the scope of facial expression classification.

Before describing these works in detail, it will be proper to look at the traget platforms which were used in these works. Looking into the examples in the listing, the target platforms are digital signal processors (DSPs), field programmable gate arrays

(FPGAs), multi-core SoC evaluation platform with ARM processor and DSPs. The diversities in computing resoruces and cabablities as well as other points of interest will be presented in consecutive sections.

The first study which will described in order to create more solid background in performance evaluation of our embedded system is [7]. This paper describes a face and facial expression recognition system on an embedded platform in order to use human-conputer interaction applications. Preferred targert platform is 150 Mhz TMS320C6711 digital signal processor (DSP) system. It has 32 MB SDRAM, 16 MB flash memory, 2-channel 0.3M Pixels Mono CMOS camera. In addition, one RS-232 port is used to communicate with a display monitor.

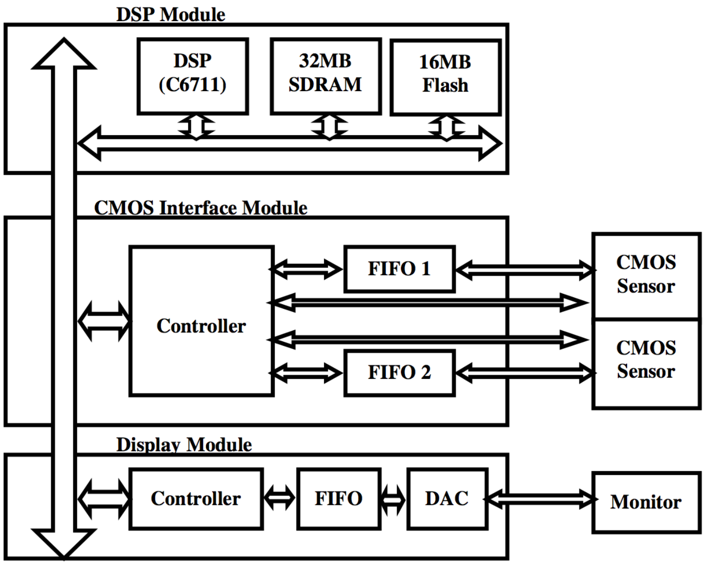

Figure 4.3: Block Diagram of Embedded Hardware system [7]

In Fig. 4.3, the general framework of embedded hardware system which is used in [7] is depicted. The system consists of three main modules: DSP module, CMOS interface module, and display module. The algorithms are tested using C/C++ and loaded within the flash memory. The training phase and learning model creation are conducted in a PC, then uploaded into flash memory.

From algorithmic perspective, this work uses an Adaboost method-based face detector which is first proposed by Viola and Jones. This method is one of the most used face detection method. However, it is relatively primitive method in comparison with the state-of-the-art in facial landmark detection and estimation. Still, it is acceptable because the implementation will be run on a embedded platform which is scarce of computing and power resources. Viola-Jones method is based on Adaboost feature selection using weak classifier and principal component analysis (PCA).

In the representation of facial expressions, geometric and illumination normalisation are first applied, and then Gabor wavelets are used as feature vectors. In classification, enhanced Fisher model (EFM) is used. The training data which is used in creation of learning model file is chosen from the Cohn-Kanade database. In performance evaluation, 500 images are selected from the Cohn-Kanade database, and half of these images are used for training purposes, the remaining are for testing. This setup is repeated 20 times and average accuracy is reported. According to the paper, the detection rate is 95 %, and facial expression classification accuracy is 93 %. Another reported performance criteria is the *processing speed of the embedded system*, and it is about **0.6** frames per second. Thus, this work is far from being a real-time system.

Another embedded example is [8]. In this paper, static face detection and emotion recognition is done using field programmable gate array (FPGA). Even if it is an FPGA implementation, there is not any related information aobut the target platform and specifications. The evaluation is done with a robot controller. The robot has a conversation with a human, then the results of classification are evaluated by three persons from different nations. If the three evaluator give the same label, the system's evaluation is accepted as true. The classification accuracy is around 70 %. However, this evaluation criteria is fairly subjective and there could be used any benchmark or setup which makes further comparisons possible.

Looking into the same paper from algorithmic approach, it is adopted facial action coding system (FACS)-based emotion classification. To see how the action unit or part based results are converted into static emotions, Fig. 4.4 is given. The methodology is composed of hue saturation luminance (HSL) conversion, skin detection and template matching, distance and orientation estimation, and emotion classification using neural networks. HSL color space is preferred, because it is easier to detect skin color in

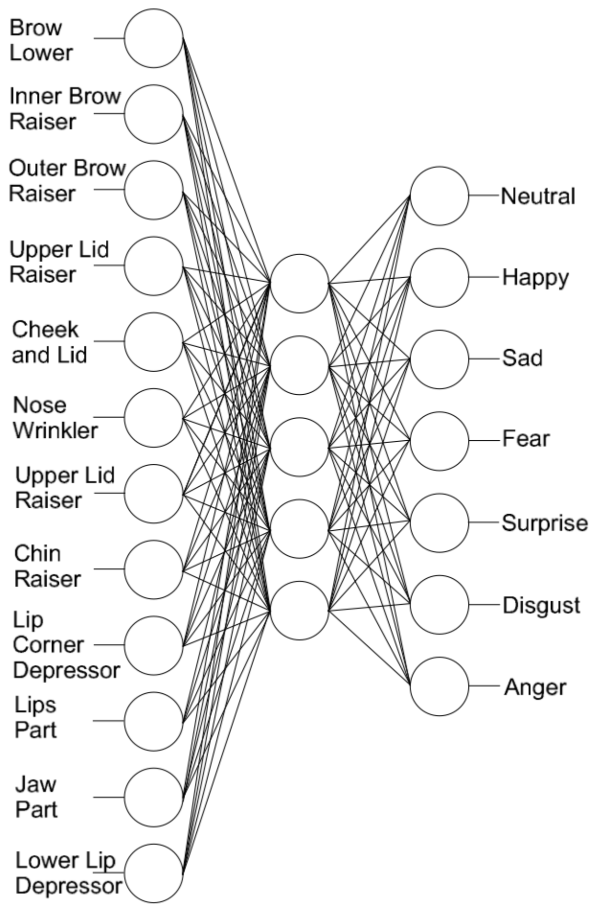

Figure 4.4: Overall Emotion Recognition Network [8]

HSL space. In feature extraction, the Gabor wavelet filters are used. In addition, it is based on FACS, so the registration is more important than a method which classifies the static emotions. In distance estimation, the position of eyes and mouth are taken as reference. In neural networks, an alternative method is developed, a structure which is named as fractional fixed point (FFP) is used instead of regular floating point calcuations. FFP neural network contains 775 inputs, each input is 31x25 pixels. A single execution of FPP neural network takes around 800ns. Unfortunately, we could not find a timing consideration of the total emotion classification task.

The third implementation example is [9]. Indeed, this study is a facial expression exaggeration system and consists of face detection, facial expression recognition, and exaggeration parts. The target platform is PAC Duo evaluation board with one ARM processor and two DSPs. It is an ideal platform for high performance and low power applications such as multimedia, and face analysis in embedded systems. In this study, two parallelised strategies are proposed and the system with ARM and PACDSP performs 4.78 fps in test frames, as well as 2.78 fps in single ARM configuration. Parallelisation strategy resulting 4.78 fps performance is given in Fig. 4.5.

Looking into the methodology of this paper, face and eye detection are done using eigenface-based face detector. In facial expression recognition, geometric features, in other words the shape and locations of facial components are used. Using optical flow to represent the facial motion, it is reported that the mouth region is very important in expressing the motion. Classification is made by a confidence factor which is up to a quadratic decision function.

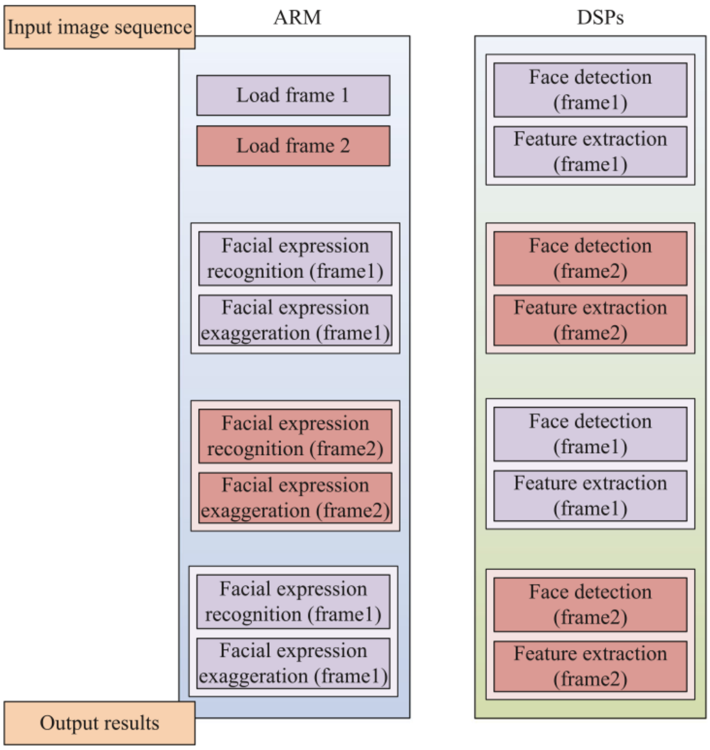

Figure 4.5: Parallelization strategy, 2nd strategy which performs 4.87 fps from [9]

The last implementation that we will consider is [31]. This implementation is done using XScale PXA270 embedded system. It is a typical SoC environment with SoC, CPU, flash ROM, RAM and USB host, and run an embedded linux operating system. The emotion classes which is used in this study are general, surprise, happy, and sadness. The performance results of these classification tasks are given as accuracy respectively as 87 %, 89 %, 91 %, and 84 % (mean accuary of all classes is 87 %).

This study is similar to the previous ones, and consists of two main parts for face detection and facial expression classification. In face detection, first skin color is detected and then binary conversion, edge detection, and noise reduction processes are applied and finally a face region is detected. If mouth region is found, it is labeled. In facial expression classification, a different approach which is based on the extension and the matter element theory is adopted. This study claims that it is a "real-time" implementation. However, it is not presented any timing results in order to evaluate this study within the requirements of real-time systems.

4.4 Comments on Implementation Performance Evaluation

In previous section, four relatively remarkable embedded implementation example which also perform emotion or facial expression classification are described. Looking into these works, we defined a more sensible and comprehensive evaluation criteria. Our findings are presented as follows:

- The target platforms vary from FPGA, to DSP or multicore architectures. The hardware based- structures such as FPGAs are not easy to update and upgrade. Because the current tendency in machine learning and computer vision develop gradually, and most of these learning algorithms needed to be implemented on hardware for these purposes.

- In terms of the facial expression recognition (FER) system, these works are similar to our implementation. However, it can be clearly seem that none of them use local descriptors such as LBP or LTP in this task.

- These works only concentrate on the embedded implementations on embedded platforms. As well as the implementation itself, the performance of the applied algorithms are important, too. Thus, we have already made a comparison among various algorithms on standardised benchmarks.

As a result, we studied these previous FER implementations, and decided to details of our design on Zynq-7000 SoC based Zedboard evaluation and development board.

4.4.1 Implementation Details

Our target platform is an ARM and FPGA-based SoC. Thus, it makes possible to implement a system using codesign methodology using both FPGA resources and ARM processor. Besides, this ARM-based module could be accelerated through special set of instructions. This instructions work with the exploitation of a single instruction multiple data (SIMD) coprocessor known as NEON.

Furthermore, we prefered the Linaro Ubuntu operating system, which has stable versions and compatible with many embedded platforms. Our system could work on another platforms which is equipped with sufficient resources and same OS.

The performance comparison of our FER system is done in previous chapters. In the extended Cohn-Kanade (CK+), the MMI facial expression, JAFFE, and static facial expressions in the wild (SFEW) databases, different feature extraction methods are used. These are local binary patterns (LBP), local ternary patterns (LTP), and Gabor filters. In computer vision literature, LBP and its variants are rather popular and widespread. We showed that its performance is better than Gabor filter, and easy to calculate in terms of time and energy resources.

LBP feature extraction :

The feature extraction and prediction using pre-learned model are two important processes in our implementation. Before explaining on-board-demo implementation, it would be proper to see the time performance of LBP feature extraction on board.

Table 4.1: LBP feature extraction time performance (sec)

Image Dimensions	Zynq-700 SoC	Computer
800x600	0.3771107	0.021721
1280x720	0.726611	0.041489
1920x1080	1.769904	0.098142
4896x2760	10.613636	0.609634

In Table 4.1, uniform local binary pattern (LBP) histograms of four test images are extracted in the neighbourhood of 8, 1. According to these results, the extraction time of LBP increases linearly. However, it is not necessary to extract feature vectors from a large area as these test images. In facial classification and emotion recognition, detected face area is downsized to at maximum 100x100 pixels patches, and the features extracted from these patches are used in classification. LBP of a 600x800 (*480,000 pixels*) test image is extracted within 0,37 second on Zynq-7000 SoC-based Zedboard.

Considering that LBP extracted patch size is *10,000* pixels at maximum, these results are promising in order to make a video-based FER system.

Our demo experiments on computer and board:

As described in previous sections, we installed Linaro Ubuntu on SD card, and our demo implementation on board is programmed using C++ and OpenCV. In Zedboard, two seperate demo application is done. One is the classification of static facial expression recognition (FER) task. Pre-trained SVM models are created on computer and copied into board's operating system.

The communication with boards OS can be done using directly ethernet connection or *ssh* to transfer test images, video, and model files. At the same time, board is connected to a wireless keyboard and mouse from USB port, and a monitor at 1280x1024 pixel resolution via HDMI.

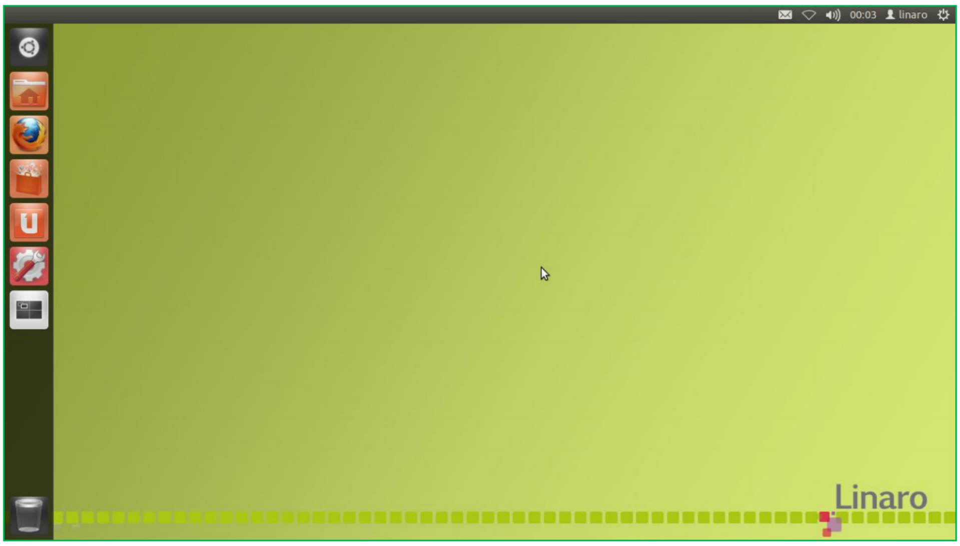

Figure 4.6: Linaro Graphical Desktop

The general framework of FER system starts with application of face detection or facial landmark localization and estimation method. After that, face region is normalised and feature vectors are made, and lastly classification is conducted using pre-trained SVM model files.

Our demostration are implemented on both Zedboard and laptop. On laptop implementation, one of the state-of-the-art methods in face alignment, *Supervised Descent Method* [32] is used. With the help of this algorithm, faces in our images are more accurately detected and aligned. Finally, we could classify more challenging data in laptop implementation.

In embedded implementation, classification model is trained using the images in CK+ database, because they reflect the nature of seven classes appearantly in comparison with the other annotated datasets.

All images and image sequences in test/demonstration phase are checked with face detector and if a face can be found, face rectangle together with a seperate eye detector's outputs are given as a reference to normalization part of the system. After registration, LBP as a feature extraction method with optimised blocking strategy is applied on images.

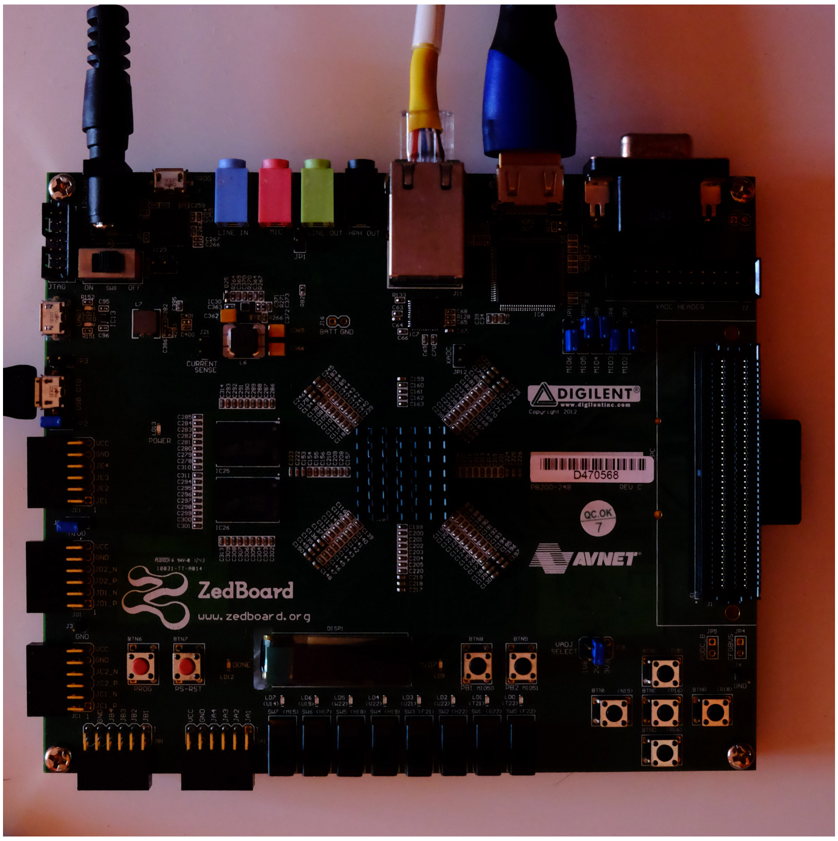

Figure 4.7: Our target platform: Zedboard Development and Evaluation Board

Embedded demonstration have two parts: One is the clasification of static facial expression on images, and the second is on selected video parts from TV series and video datasets. On image sequences, the performance of our system is approximately at 4-5 fps.

Looking into the performance of our embedded system, it is fairly good at classification of happiness, anger, surprise, and partially disgust. The performance on other classes are not very promising. The most appearant reason of this issue is the inefficient training data. For instance, CK+ images are used in training and contempt class have only 22 images.

5. CONCLUSIONS AND RECOMMENDATIONS

Facial expression (FER) and emotion recognition have a significant potential in many signal processing and computer vision tasks. However, it is not covered thoroughly, together with an embedded design and implementation. In this thesis, the theoretical background of a FER system is explained and it is implemented on embedded linux running SoC which consists of both FPGA and ARM processors. In this section, practical application of this study, our recommendation, and conclusion will be presented.

5.1 Contribution of This Study

Similar to other computer vision, facial expression recognition is a challenging task, because of various uncontrolled variable in wild environments. For instance, the implementation of a system in real-world conditions such as varying illumination, head pose, motion.

Local descriptors, especially LBP and its variants, are very successful in face analysis. There are tens of variants which are used in various applications. This study successfully compare and measure its performance on various standardised experiments and benckmarks.

In theoretical part of this study presents performance evaluation on the extended Cohn Kanade (CK+) facial expression database. In seven-class experiments, the same protocol which is described in its reference paper [1]. In this database, there are a few papers which reports the results of experiment as in the benchmark. Our LBP and SVM results are better that both the baseline in reference paper and LBP-based previous works. Our best results are in anger, happy, and surprise classes with 97.78%, 100%, and 97.59% respectively using LTP. Similary, our results in MMI and JAFFE database are very successful.

Together with LBP and LTP, Gabor filters are used in feature extraction, too. The classification accuracy of Gabor filter is not too much different from these two methods. Gabor filters are applied in 5 scales and 8 orientations. However, our final aim is to implement a novel FER system on embedded platform. Thus, time performance is one of the most important factors that should be taken into consideration in design and implementation phase. Because of this reason, we used LBP in our other experiments.

The first three database are lab-controlled datasets and we should better be sure of the performance of our descriptor and classifier in wild environment. As an example of wild data, SFEW database is used as proposed in its reference paper, following SPI protocol. Surprisingly, our LBP-based results are better than bechmark paper's fusion of LPQ and PHoG [29]. The reasons why our performance results are better are the landmark localization and affine transform before generating features, accurate and optimised blocking strategy in LBP, and working on better learning parameters in SVM.

Before starting our own design and implementation, we conducted a literature survey on embedded FER systems. Unfortunately, there are only a few studies which implements facial expression recognition on embedded platforms. First of all, the comparison and detailed analysis of these works are done, then, C++/OpenCV implementation on linux OS installed Zedboard development board is described.

The same implementation is run on both laptop and Zedboard. Laptop demonstration can use webcam video, and with the help of our own Supervised Descent Method (SDM) based face alignment, laptop demonstration makes more accurate localization and powerful in order to struggle with real-time conditions. However, embedded implementation is dependant on OpenCV's Viola-Jones cascade classifier.

Because LBP features are used in our embedded system, the time performance of LBP extraction is presented. Our final demonstration can classify static images and video parts successfully. Even if our evaluation on benchmarks are pretty good in terms of accuracy, demonstration on real-world images is very good in especially happy, anger, and surprise. The classification performance of other classes are not very succesful on especially wild data.

All in all, our embedded implementation is the first embedded facial expression recognition (FER) application (at the same time face analysis) on Zedboard and its core Zynq-7000 SoC. Our results on our demonstration showed that this architecture can be used in a prototype FER system design.

5.2 Recommendations

Our embedded design on Zedboard have promising results on discrete facial expression and emotion recognition. However, we can present our scope for future works, as follows;

1. The current implementation will be upgraded using different variants of LBP and different learning strategies such as neural networks will be reviewed.

2. The kernel of the operating system will rewrited according to the priorities and preferences of an embedded vision system. This could increase the performance of our image processing operations. At the same time, the current studies about the real-time embedded linux operation systems will be studied. In this way, we aim at developing a real-time facial expressing recognition system reliable as much as that can be used in automation applications.

3. In the performance of embedded demonstration, used images in training steps are utterly important. Unfortunately, there exists small number of annotated data in discrete emotions, action units, or dimensional models. A complete and comprehensive dataset retrieved from online images could be on the spot in order to build more robust systems in wild conditions.

REFERENCES

[1] **Lucey, P.**, **Cohn, J.**, **Kanade, T.**, **Saragih, J.**, **Ambadar, Z. and Matthews, I.** (2010). The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression, *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pp.94–101.

[2] **Xue, M.**, **Liu, W. and Li, L.** (2013). Person-independent facial expression recognition via hierarchical classification, *Intelligent Sensors, Sensor Networks and Information Processing, 2013 IEEE Eighth International Conference on*, pp.449–454.

[3] **Tian, Y.L.**, **Kanade, T. and Cohn, J.** (2001). Recognizing action units for facial expression analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ***23***(2), 97 – 115.

[4] **Kanade, T.**, **Tian, Y. and Cohn, J.F.** (2000). Comprehensive Database for Facial Expression Analysis, *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*, FG '00, IEEE Computer Society, Washington, DC, USA.

[5] **Pantic, M.**, **Valstar, M.**, **Rademaker, R. and Maat, L.** (2005). Web-based database for facial expression analysis, *Proc. IEEE Int'l Conf. Multimedia and Expo*, pp.317–321.

[6] **yue Bi, C.**, **peng Liu, Y. and fang Wang, R.** (2010). Research of key technologies for embedded Linux based on ARM, *Computer Application and System Modeling (ICCASM), 2010 International Conference on*, volume 8, pp.V8–373–V8–378.

[7] **Lee, Y.B.**, **Moon, S. and Kim, Y.G.** (2005). Face and Facial Expression Recognition with an Embedded System for Human-Robot Interaction., ***J. Tao**, **T. Tan and R.W. Picard**, editors, ACII*, volume3784 of *Lecture Notes in Computer Science*, Springer, pp.271–278.

[8] **Paul, S.J. and Dongbing, G.** (2006). Static face detection and emotion recognition with FPGA support., ***J. Andrade-Cetto**, **J.L. Ferrier**, **J.D. Pereira and J. Filipe**, editors, ICINCO-RA*, INSTICC Press, pp.390–397.

[9] **Su, T.F.**, **Duan, C.H.**, **Wang, S.F.**, **Lee, Y. and Lai, S.H.** (2014). Automatic Facial Expression Exaggeration System with Parallelized Implementation on a Multi-Core Embedded Computing Platform, *Signal Processing Systems*, ***75***(2), 155–168.

[10] **Ekman, P.**, **Friesen, W. and Hager, J.** (2002). *Facial Action Coding System (FACS): Manual*, A Human Face, Salt Lake City (USA).

[11] **Lanitis, A.**, **Taylor, C.J. and Cootes, T.F.** (1997). Automatic interpretation and coding of face images using flexible models, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ***19***, 743–756.

[12] **Kapoor, A.**, **Qi, Y.A. and Picard, R.W.** (2003). Fully Automatic Upper Facial Action Recognition, *AMFG*, IEEE Computer Society, pp.195–202.

[13] **Valstar, M.F. and Pantic, M.** (2006). Fully automatic facial action unit detection and temporal analysis, *Proceedings of IEEE Int'l Conf. Computer Vision and Pattern Recognition (CVPR'06)*, volume 3, New York, USA, p.149.

[14] **Lyons, M.J.**, **Budynek, J. and Akamatsu, S.** (1999). Automatic Classification of Single Facial Images, *IEEE Trans. Pattern Anal. Mach. Intell.*, ***21***(12), 1357–1362.

[15] **Tian, Y.L.** (2004). Evaluation of Face Resolution for Expression Analysis, *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, ***5***.

[16] **Ojala, T.**, **Pietikäinen, M. and Harwood, D.** (1996). A comparative study of texture measures with classification based on featured distributions., *Pattern Recognition*, ***29***(1), 51–59.

[17] **Ojala, T.**, **Pietikäinen, M. and Mäenpää, T.** (2002). Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns, *IEEE Trans. Pattern Anal. Mach. Intell.*, ***24***(7), 971–987.

[18] **Lyons, M.**, **Akamatsu, S.**, **Kamachi, M. and Gyoba, J.** (1998). Coding Facial Expressions with Gabor Wavelets, *Proceedings of the 3rd. International Conference on Face & Gesture Recognition*, FG '98, IEEE Computer Society, Washington, DC, USA.

[19] **Bartlett, M.S.**, **Hager, J.C.**, **Ekman, P. and Sejnowski, T.J.**, (1999), Measuring facial expressions by computer image analysis.

[20] **Littlewort, G.**, **Whitehill, J.**, **Wu, T.**, **Fasel, I.R.**, **Frank, M.G.**, **Movellan, J.R. and Bartlett, M.S.** (2011). The computer expression recognition toolbox (CERT), *FG*, pp.298–305.

[21] **Rahman, Z. and Woodell, G.A.** (1997). A multiscale retinex for bridging the gap between color images and the human observation of scenes, *IEEE Transactions on Image Processing*, ***6***, 965–976.

[22] **Chen, T.**, **Yin, W.**, **Zhou, X.S.**, **Comaniciu, D. and Huang, T.S.** (2006). Total Variation Models for Variable Lighting Face Recognition, *IEEE Trans. Pattern Anal. Mach. Intell.*, ***28***(9), 1519–1524.

[23] **Gross, R. and Brajovic, V.** (2003). An Image Preprocessing Algorithm for Illumination Invariant Face Recognition, *4th International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA)*, Springer.

[24] **Tan, X. and Triggs, B.** (2010). Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions, *Trans. Img. Proc.*, ***19***(6), 1635–1650.

[25] **Pietikainen, M.**, **Hadid, A.**, **Zhao, G. and Ahonen, T.** (2011). *Computer Vision Using Local Binary Patterns*, Computational imaging and vision, Springer Verlag, London.

[26] **Daugman, J.G.** (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters, *J. Opt. Soc. Am. A*, ***2***(7), 1160–1169.

[27] **Cortes, C. and Vapnik, V.** (1995). Support-Vector Networks, *Machine Learning*, ***20***(3), 273–297.

[28] **Valstar, M.F. and Pantic, M.** (2010). Induced Disgust, Happiness and Surprise: an Addition to the MMI Facial Expression Database, *Proceedings of Int'l Conf. Language Resources and Evaluation, Workshop on EMOTION*, Malta, pp.65–70.

[29] **Dhall, A.**, **Goecke, R.**, **Lucey, S. and Gedeon, T.** (2011). Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark., *ICCV Workshops*, IEEE, pp.2106–2112.

[30] **Richter, M.**, (2012), Facial Expression Classification on Web Images.

[31] **Neng-Sheng, P. and Shih-Ping, C.** (2011). An embedded system for real-time facial expression recognition based on the extension theory, *Computers and Mathematics with Applications*, ***61***(8), 2101–2106.

[32] **Xiong, X. and De la Torre, F.** (2013). Supervised Descent Method and its Applications to Face Alignment, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

CURRICULUM VITAE

Name Surname: Ömer Sümer

Place and Date of Birth: İzmir, December 30, 1988

Adress: İstanbul

E-Mail: omer.sumer@outlook.com

B.Sc.: Deniz Harp Okulu (Turkish Naval Academy), Tuzla - İstanbul