Adaptive Robust Local Complete Pattern For Facial Expression Recognition

Authors

Al Shahriar Rubel (144434)

Adib Ahsan Chowdhury (144446)

Supervisor

Dr. Md. Hasanul Kabir

Professor

Department of Computer Science and Engineering (CSE)

Islamic University of Technology (IUT)

A thesis submitted to the Department of CSE

in partial fulfillment of the requirements for the degree of

B.Sc. Engineering in CSE

October - 2018

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by Al Shahriar Rubel and Adib Ahsan Chowdhury under the supervision of Dr. Md. Hasanul Kabir, Professor, Department of Computer Science and Engineering, Islamic University of Technology (IUT), Dhaka, Bangladesh. It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Authors:

_____

Al Shahriar Rubel
Student ID: 144434

_____

Adib Ahsan Chowdhury
Student ID: 144446

Supervisor:

_____

Dr. Md. Hasanul Kabir
Professor,
Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)

Abstract

An effective and robust face descriptor is an essential component for a good facial expression recognition system. Many popular appearance-based methods such as local binary pattern (LBP), local directional pattern (LDP) and local ternary pattern (LTP) have been proposed to serve this purpose and have been proven both accurate and efficient. During the last few years, many researchers have been providing significant effort and ideas to improve these methods. In this research work, we present a new face descriptor, Adaptive Robust Local Complete Pattern (ARLCP). ARLCP effectively encodes significant information of emotion-related features by using the sign, magnitude and directional information of edge response that is more robust to noise and illumination variation. In this histogram-based approach, obtained feature image is divided into several regions, histogram of each region is computed independently and all histograms are concatenated to generate a final feature vector. We have experimented our method on several datasets using cross-validation schemes to evaluate the performance. From those experiments, it is evident that our method(ARLCP) provides better accuracy in facial expression recognition.

Acknowledgements

First of all, we are really grateful the Almighty Allah for everything that happened in our life. Without His help, blessings and mercy, nothing would have been possible. “All the thanks and praises to Allah”.

We would like to thank our supervisor, **Dr. Md. Hasanul Kabir**, Professor, Department of Computer Science and Engineering, Islamic University of Techonology, Gazipur, Bangladesh for his kind advice, support and counsel. Dr. Kabir has been an instrumental to this work and our careers. He taught us to work in research sectors and how to generate new ideas for solving problems. He also helped us to think out of the box to generate ideas like Adaptive Robust Local Complete Pattern(ARLCP). Without his helps and visions, the completion of ARLCP would not be possible. Sir may the Almighty Allah reward you abundantly.

Also, it is our pleasure to get the cooperation and coordination from our honorable Head of the Department, **Professor Dr. Muhammad Mahbub Alam** during various phases of the work. The faculty members of CSE departments helped us by providing a helpful set of eyes and ears when problems arose. Moreover, the lab assistants of CSE department are the unsung heroes of our work who helped us by providing technical and instrumental supports.

Finally, we are extremely grateful to all my friends and family for their unconditional support. This work would have never been completed without the consistent support and encouragement from them throughout our undergraduate program.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In 1975, when an engineer at Eastman Kodak named Steven Sasson invented the first ever digital camera, no one could have imagined how far this technology would go in such little time[1]. With the advancements of computer science & the refinements of digital imaging technologies, Digital Image Processing(DIP) has become a rising section of modern technologies. Starting as a subcategory of digital signal processing, nowadays DIP has so much more applications that often it is considered to be a very powerful aspect of technology. That is why back in 2011, the Economic Times of India defines it as "the next big thing" to change the world[2].

1.1 Digital Image Processing in Facial Feature Extraction

Digital Image Processing(DIP) so far has been involved in a lot of cases. One of the biggest use of DIP is in the facial feature extractions. Facial feature extraction is the process of extracting face component features like eyes, nose, mouth, etc from human face image. Facial feature extraction is very much important for the initialization of processing techniques like face tracking, facial expression recognition and face recognition[3].

With the advancement of DIP in facial feature extractions, once mere impossible works are now very much possible to do. Big companies like Facebook, Google, Apple are now investing big money and efforts to improve in these sectors. Facebook uses these facial features to help the users to find their own photos and videos. Law enforcement agencies often use these technologies to detect the perpetrators from crime scenes. Even some of the recent works can help to identify genetic conditions through facial recognitions[4].

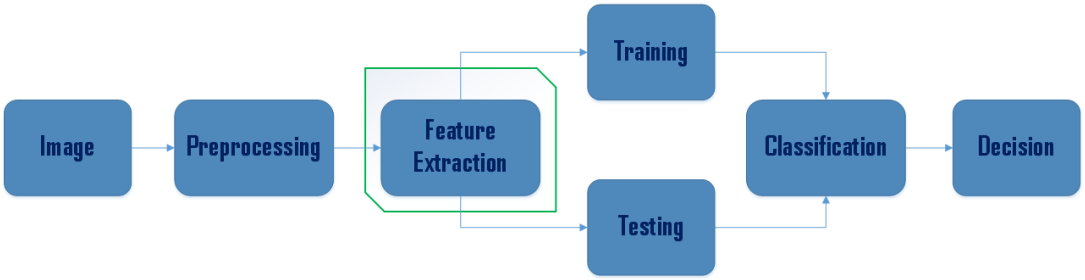

Figure 1.1: Overview of a generic facial recognition System

All these facts suggest that automatic recognition and analysis of human face allow many interesting applications in Biometrics, Human-Computer Interaction, and Security. From early 90's to till date, many researchers are working to find newer & effective ways to derive features from human faces. As a result, more & more ways of finding effective ways to extract features are coming in our way.

Deriving an efficient and effective feature representation is the fundamental component for any successful facial recognition system. So, here our ongoing research aims to increase the robustness of the underlying feature descriptor against different factors.

1.2 Problem Statement

Here we want to create a system for facial feature extraction for expression detection. From the Figure 1.1, we can see that extracting information is the most important part of any facial feature extraction system. This face descriptor should be robust in uncontrolled environment like noise, pose variation, illumination variation, aging, occlusion etc.[5]. So, designing a robust facial feature extractor is a critical task. Our aim is to create such a facial feature extractor which overcomes all of these obstacles.

Here is an image of all basic seven expressions in facial feature extractions.

A good feature representation should have the ability to recognize all these

Figure 1.2: Different facial expressions from - (a) CK+ dataset and (b) JAFFE dataset

expressions in all given environment.

1.3 Research Challenges

All of the research works begins with some challenges. And for our case, it is also the same. Along with the data problems, there are also some issues which are appeared to be the research challenges such as-

- variations in illumination
- occlusion
- facial alignments
- different facial poses
- aging

1.3.1 Variations in illumination, pose, alignment

The first of the problems have come from the variation of illumination. Different illumination does not change the facial features, but it definitely changes the pixel values. And in digital image processing, everything depends on the pixel values.

The alignment and pose of image subject can also affect the outcome. Overcoming these variations of illumination, pose and alignment is a major obstacle.

1.3.2 Occlusion

The meaning of the word 'Occlusion' means obstacle. Often there are some obstacles presented in the image. These sort of obstacles should be taken care of.

1.3.3 Aging

Age of the subject can also create variation in image data.

Human faces are non-rigid, dynamic objects with a large diversity in shape, color and texture. So a neat, robust, distinctive and effective descriptor is needed to extract facial features from a human face.

1.4 Objectives

Our thesis work focuses on achieving a Facial Feature Descriptor using directional micro-patterns which can provide better accuracy than the existing ones. Our thesis work will satisfy the following criteria :

- Robust against illumination variations and random noise.
- Applicable in different face related problem domains, such as facial expression analysis, face recognition, gender classification, age estimation and so on.
- Achieve superior performance in unconstrained environment than the existing state-of-the-art techniques.

In the following chapter we will show the literates that have been analyzed and implemented by us.

1.5 Contribution

In this field of research, we have introduced a new facial feature extraction method which we are calling Adaptive Robust Local Complete Pattern (ARLCP).

In this method, a new local pattern for effective facial feature representation using directional edge responses[3] is introduced, which includes the description of the basic ARLCP operator and how to construct facial feature descriptor based on ARLCP. We effectively explore the advantage of edge response by incorporating the sign, magnitude and directional information that leads to more robust feature representation.

1.6 Organization of the Thesis

This thesis book has five chapters. Chapter 1 introduced about the facial feature representation system. Also the problem domain and research challenges of the system are also stated here. Chapter 2 gives us a overview of Related work on this facial feature representation where we can get the idea of how far the research has gone in this field. Chapter 3 describes our proposed face descriptor named Adaptive Robust Local Complete Pattern(ARLCP).Chapter 4 shows the analysis of different algorithms and comparative performance of our proposed feature descriptor. And finally in Chapter 5 we have referred to conclusion with some future plan for better facial feature representation. References are added at the end of this book.

Chapter 2

Literature Review

To get some ideas about our field of work, we have gone through many research papers. In this chapter, we first present a discussion on different geometric and appearance-based facial feature representation, which is followed by a review on different local appearance-based methods. From there, we talk about various ideas and implementations that finally leads us to our final proposed method.

2.1 Facial Feature Representation Methods

In last two decades, many different methods have been proposed in the field of facial feature representation. Face recognition, expression detection and many other sectors are getting benefited from these representation methods. Here we divide these methods in two basic types - $i.$ Geometric feature-based methods and $ii.$ Appearance-based methods.

2.1.1 Geometric feature-based methods

Geometric feature learning is a technique combining machine learning and computer vision to solve visual tasks[6]. The feature vector is formed based on the geometric relationships such as positions, angles or distances between different facial components (eyes, ears, nose etc.). Many of the primitive facial recognition systems are mainly based on geometric feature-based methods.

In the beginning, facial action coding system (FACS)[7] was a popular geometric feature-based method which represents facial expression with the help of a set of action units (AU). Physical behavior of a specific facial muscle is represented by each action unit. Then, a feature extraction method based on the geometric positions of 34 manually selected fiducial points is proposed

by Zhang[8].A similar representation was adopted by Guo and Dyer[9]. In that method, they employed linear programming in order to perform simultaneous feature selection and classifier training. In recent works in[10] ,it has been found that geometric features provide similar or better performance than appearance-based methods in action unit recognition.

For face recognition, elastic bunch graph match (EBGM)[11] algorithm is the most commonly-used geometric feature-based method. In this method, human faces are represented as graphs, where nodes are positioned on fiducial points and edges are labeled using distance vectors. A set of 16 geometric features for designing a gender classification model is used by Brunelli and Poggio[12] in which the features were extracted with HyperBF networks. Furthermore, a radial basis function (RBF) network and a perceptron is experimented by Abdi et al.[13]. Here they achieved good classification result for both pixel-based inputs and measurement-based inputs (geometric features). However, the effectiveness of geometric methods is heavily dependent on the accurate detection of facial components, which is a difficult task in changing and unconstrained environment, thus making geometric methods difficult to accommodate in many scenarios.

2.1.2 Appearance Based Methods

In appearance-based methods, the facial appearance is extracted by applying image filter or filter bank on the whole face image or some specific facial regions. Principal component analysis (PCA)[14], independent component analysis (ICA)[15, 16], Gabor wavelets[17, 18] and more recent enhanced ICA (EICA)[19] are the commonly-used appearance-based methods for facial expression detections. Here, PCA is a global feature extraction method and the whole facial image is taken into account during feature vector generation. An optimal linear transformation from the original image space is provided to an orthogonal eigenspace with reduced dimensionality in the sense of least mean squared reconstruction error[20].

The “Eigenface" method was introduced by Turk and Pentland [21]. Later,

linear discriminant analysis (LDA)[22], 2D PCA [23], local features analysis (LFA)[24], and dynamic link architecture (DLA) [25] methods were also investigated for feature representation in face recognition systems. But their performance deteriorate in changing environment[26]. Donato et al.[27] presented a comprehensive analysis of different techniques for facial action recognition, which included PCA, ICA, local feature analysis (LFA), Gabor wavelets, and local principal components (PCs). Among these techniques, ICA and Gabor wavelets provided the best recognition rate.

In recent days, local appearance face descriptors based on local binary pattern (LBP) [28, 29] and its variants[30] start to gain popularity. Local binary pattern is a simple, yet effective local texture description technique, which is computationally efficient and robust against non-monotonic illumination variation. However, the LBP method performs weakly under the presence of large illumination variation and random noise [31], since a little variation in the gray level can easily change the LBP code. Later, local ternary pattern (LTP) [31] was introduced to increase the robustness of LBP in uniform and near-uniform regions by adding an extra intensity discrimination level and extending the binary LBP value to a ternary code. Another method named local directional pattern (LDP) [32] employed a different texture encoding approach, where directional edge response values around a position is used instead of gray levels. The motivation was to exploit more stable edge response values instead of gray levels in order to encode the local texture, which increases the robustness of the underlying feature descriptor.

2.2 Local Texture Operators

In this section, we present a review on some widely-used local pattern-based facial feature descriptors. Here we divide these papers in three categories-

1. **Basic Patterns:**

   (a) Local Binary Pattern

(b) Local Binary Count

(c) Local Ternary Pattern

2. **Complete Patterns:**

   (a) Complete Local Binary Pattern

   (b) Complete Local Binary Count

   (c) Complete Local Ternary Pattern

3. **Adaptive Patterns:**

   (a) Adaptive Mean Binary Pattern

   (b) Adaptive Robust Binary Count

   (c) Directional Age Primitive Pattern

2.3 Basic Patterns:

For our works of study, the Local Binary Pattern(LBP) and Local Ternary Patterns are considered to be the basic patterns. These methods are pretty straightforward and the evolution of facial features extractions began with these methods.

2.3.1 Local Binary Pattern(LBP)

Local Binary Pattern(LBP) is considered to be one of the oldest and most used algorithms for facial feature extractions. This ground-breaking approach was introduced in 1996 by Ojala et al. [33] which forms labels for the image pixels by thresholding the 3 x 3 neighborhood of each pixel with the center value and considering the result as a binary number. The method is widely used for finding the texture pattern which helps to find the similarity between two images.

In LBP method, some local neighbor values, P is taken around each pixel and generates a P-bit binary code by thresholding the intensity value of the neighbour pixels with respect to the intensity of the center pixel. The binary code of LBP is generated using equation 2.1.

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p; \tag{2.1}$$

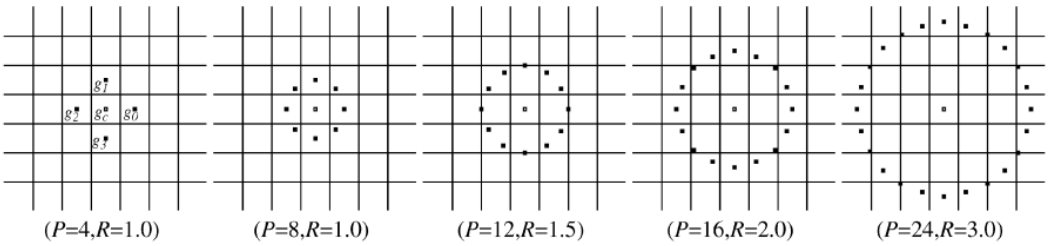

Figure 2.1: Circularly symmetric neighbor sets for different (P, R)

Here, g_c denotes the value of a center pixel and g_p defines the values of the local neighborhood pixels. P is the number of the neighbors and R is the radius of the neighbors as it is shown in figure 2.1. If the function generates a value which is greater or equal to 1, it assigns a binary code "1" for the pixel, otherwise that pixel is assigned with the value 0 as it is stated in the equation 2.2.

$$s(x) = \begin{cases} 1, & \text{if } x \geqslant 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

The generation of P-bit code is shown here in the figure 2.2 where $P = 8$ and it takes a 3×3 region for applying the LBP method.

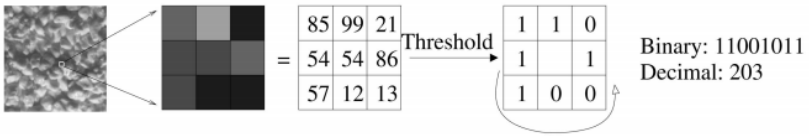

Figure 2.2: The Basic LBP operator

Limitations of LBP

LBP is quick, easy and simple. But it is an intensity based operator. So noise can cause changes in the LBP code. As a result it will affect the performance and accuracy of the method. Generally, the presence of noise results in the decrease of accuracy and performance.

2.3.2 Local Binary Count

Local Binary Count (LBC)[34] is a variant of LBP. In the original LBP and its variants, each pixel in the local neighbor set is turned to binary form by comparing it with the central pixels and then are encoded to form the local binary patterns. But in case of LBC, only the number of value 1's in the binary neighbor sets are counted instead of encoding them. The following figure shows the working principles of LBC.

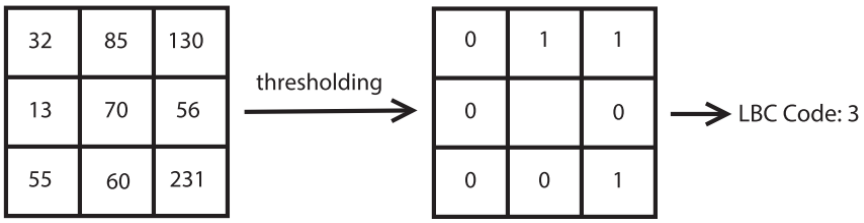

Figure 2.3: Illustration of LBC (P = 8, R = 1)

Here, the number of value 1's is four in the binary neighbor set, thus the LBC code of the central pixel is also three. We can define the computing process for the LBC as follows:

$$LBC_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c); \qquad s(x) = \begin{cases} 1, & \text{if } x \geqslant 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

Like LBP, here also g_c defines the center pixel intensity and g_p defines the values of the local neighborhood pixels. P is the number of the neighbors and R is the radius of the local neighborhood. LBP and LBC mostly differed in the use of pixel values. In case of LBP, it uses the generated binary number to encode local patterns while the LBC merely counts the number of value 1's in local neighbor set. Usually, LBP is mainly used for finding a structural information characterized by various patterns in the local neighbor while LBC focuses on finding the count of the pixels which are greater than the center one. Thus pixel positions can cause some effects in LBP but in case LBC, only the count of pixels that matters.

In statistical texture analysis methods, the repeats for a large number of lo-

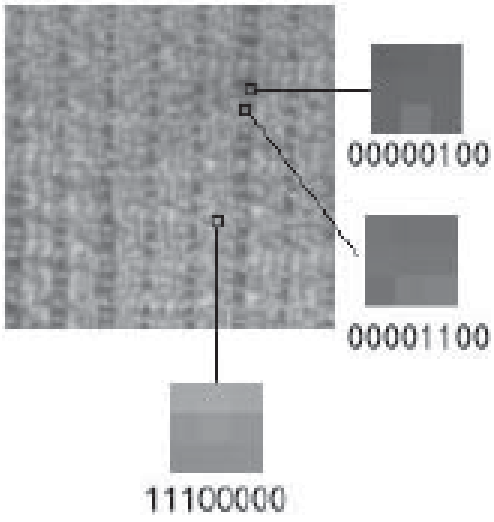

Figure 2.4: Schematic diagram of macroscopic textural structure that is quite different from the micro-structure.

cal microcosmic patterns are regarded to be the microscopic structures. The whole structure can be characterized using the statistics of the selected local microcosmic patterns which are quite different from the macroscopic textural structure. In the figure 2.4, the micro-structures, “00000100” and “00001100”, may be contained in a macroscopic “line" in texture image, and a micro-structure “line" which is “11100000” may be a “spot" in the image. That is a macroscopic textural structure in the real image which may consist of many different local binary micro-structures. Different macroscopic textural structures may have similar micro-structures, but can different in frequencies of occurrence. According to Ojala et al. in 2002[35], the local binary pattern can characterize the local texture effectively by detecting “micro-structure" . The LBP characterizes the distribution of local pixels effectively by using the local binary encoding but cannot detect the frequency of occurrences. This is where LBC comes in handy as it can distinguish the different distributions of local pixels. Thus the LBC features can also represent the macroscopic textural structures.

Limitations of LBC

LBC depends of the count of the pixels. As a result, it can generate same codes for different positional patterns which may cause errors in local structural patterns.

2.3.3 Local Ternary Pattern :

Local binary pattern (LBP) is very sensitive to noise. Local ternary pattern (LTP) partially solves this problem by encoding the small pixel difference into a third state. Local Ternary Pattern (Tan and Triggs, 2010)[36] introduces one additional discrimination level than LBP. the pixel difference between the center pixel and the neighboring pixel was encoded into a trinary code.

$$LTP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p; \qquad s(x) = \begin{cases} 1, & x \geqslant t \\ 0, & -t < x < t \\ -1, & x < -t \end{cases} \tag{2.4}$$

Here in the equation 2.4, g_c defines the center pixel intensity and g_p defines the values of the local neighborhood pixels. P is the number of the neighbors and R is the radius of the local neighborhood. "t" here defines a user defined threshold. Thus this 3-valued code is more resistant to noise, but it is no longer strictly invariant to gray level transformations. Here is an illustration of LTP where the threshold is set to 5[37].

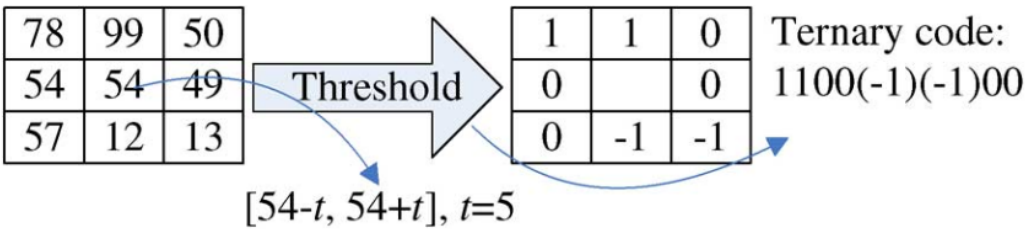

Figure 2.5: Illustration of the basic LTP operator

When using in equation 2.4 a 3^n valued code is generated but the uniform pattern argument are needed to be applied here. Thus it is desirable to reduce its dimensionality[38]. So, the experiment below uses a coding scheme that splits each ternary pattern into its positive and negative halves as illustrated in figure 2.6 which are subsequently treated with two different LBP descriptors for which separate histograms and similarity metrics are computed, combining

the results only at the end of the computation[31].

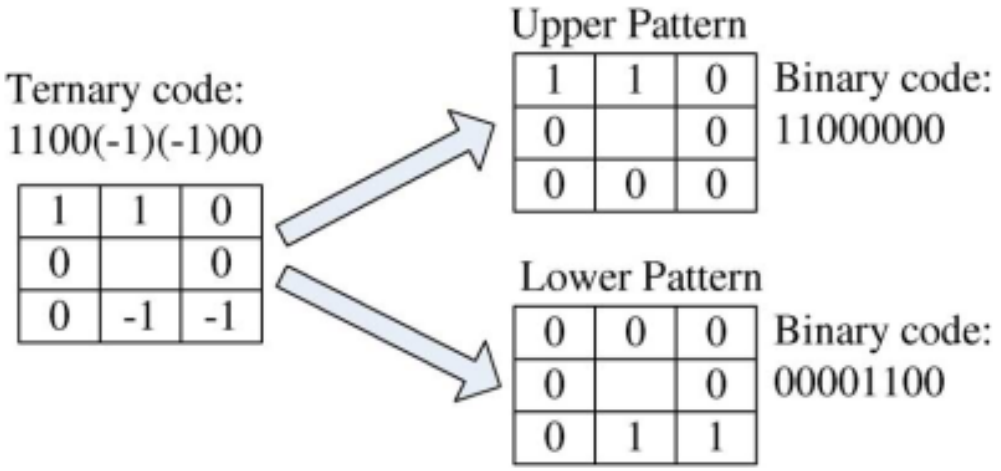

Figure 2.6: Illustration of the basic LTP operator

Limitations of LTP

Though LTP performs better than LBP, it is still susceptible to noise and illumination variations as it employs simply the gray level values.

2.4 Complete Patterns:

Complete patterns are a fusion of a few image components. Instead of the total intensity of the image, here the value comparison of sign, magnitude of difference between center and neighbors and the center intensity value is used. All these values should provide more robustness to the descriptors.

2.4.1 Complete Local Binary Pattern:

The completed LBP (CLBP) descriptor was proposed by Guo et al.[39] in order to improve the performance of LBP descriptor. Simple LBP techniques are good for showing local image features. Also there is a magnitude component in images which may contribute additional discriminant information if it is properly used. Moreover, there are a lot of potential uses of the center pixel value of the picture[40]. So, CLBP combines all these three components to make a more robust descriptor. From figure 2.7, we can see that, the local difference in

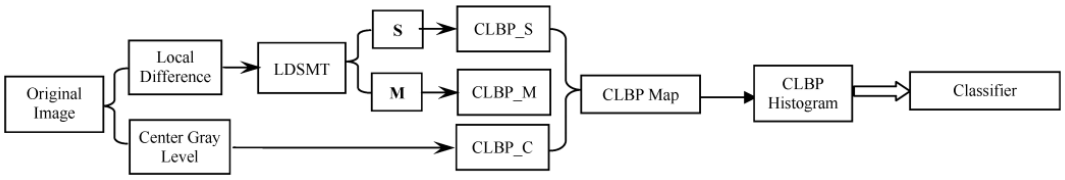

Figure 2.7: Framework of CLBP

the image is converted into Sign(S) and Magnitude(M) values which are here denoted as CLBP_S and CLBC_M. The center pixel(C) is also used which is denoted here by CLBP_C. Then all these three codes are combined to create CLBP histogram which can be used for texture classifications.

Here CLBP_S is created using the same equations from the LBP method.

$$CLBP\_S_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p; \qquad s(x) = \begin{cases} 1, & x \geqslant 0 \\ 0, & \text{Otherwise} \end{cases} \tag{2.5}$$

But as the magnitude components are not binary like "1" or "-1", rather they are continuous values. So, in order to code for magnitude values, the CLBP_M operator is calculated using:

$$CLBP\_M_{P,R} = \sum_{p=0}^{p-1} t(m_p, c)2^p; \qquad t(x, c) = \begin{cases} 1, & x \geqslant c \\ 0, & \text{Otherwise} \end{cases} \tag{2.6}$$

in this equation, c is a threshold to be determined adaptively as the mean value of the whole image. m_p is the magnitude of difference between center & neighbor pixels. Both of these CLBP_S and CLBP_M produce binary string so that they can be conveniently used together for pattern classification using concatenation.

The third operator is the CLBP_Center which is defined here by :

$$CLBP\_C_{P,R} = t(g_c, c_1) \tag{2.7}$$

where t is defined in equation 2.6 and c_1 is set as the average gray level of the whole image. All of these three components are combined to create a histogram which works as the feature descriptor.

Limitations of CLBP

Though CLBP is better than LBP, it performs weakly under the presence of random noise and large illumination variation as it uses the direct intensity values like LBP.

2.4.2 Complete Local Binary Count(CLBC)

Similar to CLBP, LBC is extended to completed LBC (CLBC) which is proposed in[41]. This procedure is also devided into three operators : CLBC-Sign(CLBC_S), CLBC-Magnitude(CLBC_M) and CLBC-Center(CLBC_C). These operators can be combined into joint or hybrid distributions and they are used for rotation invariant texture classification.

Here CLBC_S is basically the basic LBC descriptor.

$$CLBC\_S_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c); \qquad s(x) = \begin{cases} 1, & x \geqslant 0 \\ 0, & \text{Otherwise} \end{cases} \tag{2.8}$$

the meaning of g_c, g_p, P & R is stated in the equation 2.3. The CLBC_M counts how many neighbors have comparatively much higher intensity than the center pixel which is used to extract local intensity differences.

$$CLBC\_M_{P,R} = \sum_{p=0}^{P-1} t(m_p, c); \qquad t(x, c) = \begin{cases} 1, & x \geqslant c \\ 0, & \text{Otherwise} \end{cases} \tag{2.9}$$

the values of m_p, c are stated in the equation 2.6. Also according to [31], the center pixel can be used to express the local gray level in the image which is expressed in CLBC_C.

$$CLBC\_C_{P,R} = t(g_c, c_1) \tag{2.10}$$

As we can see, it is pretty similar to equation no. 2.7. Similarly using the count of the pixels, three different histograms are generated which are finally used as the feature descriptor.Thus CLBC discards the structural information from LBP operator and only concentrate on local binary gray-scale difference.

Limitations of CLBC

It may generate same codes for different pixels as it only counts the total number of 1s rather than the position like LBC.

2.4.3 Complete Local Ternary Pattern(CLTP)

Complete Local Ternary Pattern(CLTP) (T. H. Raseen & B. E. Khoo, 2013)[42] extends the concept of LTP & its framework is similar to CLBP. It capitalizes the ternary pattern system of LTP and thus for local difference of the image, the value of sign and magnitudes are decomposed into two complementary components as follows :

$$s_p^{upper} = s(i_p - (i_c + t)), \qquad s_p^{lower} = s(i_p - (i_c - t))$$

$$m_p^{upper} = |i_p - (i_c + t)| \qquad m_p^{lower} = |i_p - (i_c - t)|$$

Here i_c defines the intensity of the center pixel where i_p shows the intensity value of p_{th} pixel. t is the user defined threshold for the image. For CLTP_Sign, the s_p^upper and s_p^lower are used to build the $CLTP\_S_{P,R}^{upper}$ and $CLTP\_S_{P,R}^{lower}$ respectively.

$$CLTP\_S_{P,R}^{upper} = \sum_{p=0}^{P-1} s(i_p - (i_c + t)2^p; \qquad s_p^{upper} = \begin{cases} 1, & i_p \geqslant i_c + t \\ 0, & \text{Otherwise} \end{cases} \tag{2.11}$$

$$CLTP\_S_{P,R}^{lower} = \sum_{p=0}^{P-1} s(i_p - (i_c - t)2^p; \qquad s_p^{lower} = \begin{cases} 1, & i_p < i_c - t \\ 0, & \text{Otherwise} \end{cases} \tag{2.12}$$

The $CLTP\_S_{P,R}$ is the concatenation of the $CLTP\_S_{P,R}^{upper}$ and the $CLTP\_S_{P,R}^{lower}$ which is as follows:

$$CLTP\_S_{P,R} = [CLTP\_S_{P,R}^{upper}, CLTP\_S_{P,R}^{lower}] \tag{2.13}$$

In the similar way , $CLTP\_M_{P,R}$ is built using the two magnitudes of m_p^{upper} and m_p^{lower} as follows:

$$CLTP\_M_{P,R}^{upper} = \sum_{p=0}^{P-1} t(m_p^{upper}, c)2^p; \qquad t(m_p^{upper}, c) = \begin{cases} 1, & |i_p - (i_c + t) \geqslant c \\ 0, & \text{Otherwise} \end{cases} \tag{2.14}$$

$$CLTP\_M_{P,R}^{lower} = \sum_{p=0}^{P-1} t(m_p^{lower}, c)2^p; \qquad t(m_p^{lower}, c) = \begin{cases} 1, & |i_p - (i_c - t) \geqslant c \\ 0, & \text{Otherwise} \end{cases} \tag{2.15}$$

$$CLTP\_M_{P,R} = [CLTP\_M_{P,R}^{upper}, CLTP\_M_{P,R}^{lower}] \tag{2.16}$$

Here i_c ,i_p P, R and t is defined before. Finally, the third component which involved the center pixel value is mathematically described as follows:

$$CLTP\_C_{P,R}^{upper} = t(i_c^{upper}, c_I), CLTP\_C_{P,R}^{lower} = t(i_c^{lower}, c_I), \tag{2.17}$$

where $i_c^{upper} = i_c + t$, $i_c^{lower} = i_c - t$ and c_I is the average gray level of the whole image. After getting all these three histograms, these are combined to create the feature vectors in CLTP.

Limitations of CLTP

As it uses the same concept of LTP, it is susceptible to noise and illumination variations.

2.5 Adaptive Patterns

In this category, methods with adaptive thresholding are only going to be discussed.

2.5.1 Adaptive Median Binary Pattern(AMBP)

Adaptive Median Binary Pattern (AMBP) is variation of Median Binary Patterns(MBP) operators which uses an adaptive thresholding. The MBP operator maps from the intensity space to create a binary pattern by thresholding the pixels against their median value within a neighborhood. According to [43], the MBP at pixel (i, j) is given in figure 2.8 using

$$MBP(i,j) = \sum_{k=0}^{L-1} 2^k H(b_k - \tau); \qquad H(b) = \begin{cases} 1, & b \geqslant 0 \\ 0, & \text{Otherwise} \end{cases} \tag{2.18}$$

Here b is the intensity of b^{th} pixel and H is the Heaviside unit step function which is also referred to be binary thresholding function. L is the patch size and τ is considered to be the constraining threshold which is the median value in case of MBPs.

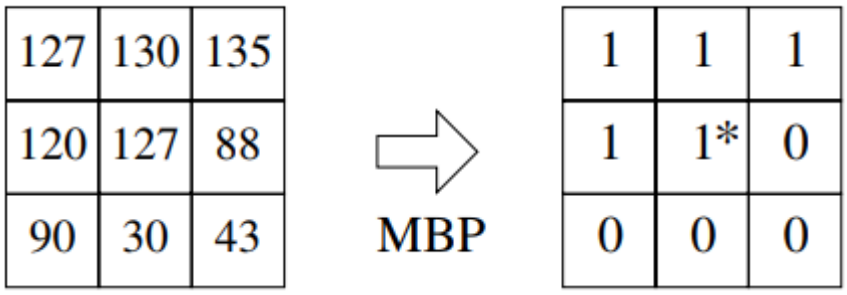

Figure 2.8: MBP operator for 3X3 patches

In figure 2.8, an MBP operator for a 3×3 patch (i.e. L = 9) with a median value of 120 (i.e. $\tau = 120$) is used. Output binary pattern of MBP = 100011110_2 = 286. Ordering of bits in the binary pattern uses b_0 as the least significant bit which is shown in the following figure 2.9 :

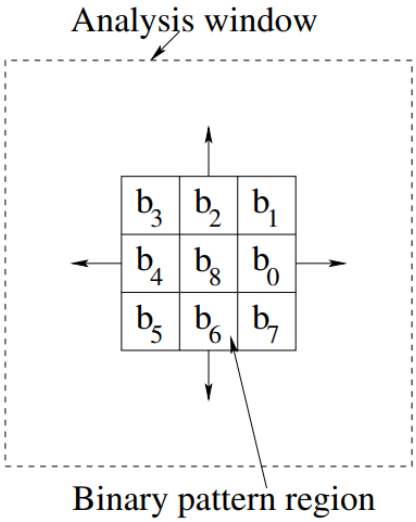

Figure 2.9: Illustration of AMBP terminology

However, a small windows can miss a lot of information and also a larger window can be sometimes bad for performances. Thats is why, AMBP considers larger and adaptive analysis windows to compute optimal local threshold.

$$AMBP(P,R) = \sum_{p=0}^{P-1} 2^k s(g_p - \tau) \qquad s(x) = \begin{cases} 1, & x \geqslant 0 \\ 0, & \text{Otherwise} \end{cases} \tag{2.19}$$

Here in the equation 2.19, P is defined as the patch size. The values of g_p, τ and $s(x)$ are described in previous equations. In the AMBP algorithm from algorithm 1 where, S is the square region around a pixel, k_{max} is the maximum analysis window. The median, minimum and maximum value inside S is stated by Z_{med},Z_{min} and Z_{max} respectively.

Limitations

AMBP is sensitive to noise as it uses the intensity values like LBP.

2.5.2 Adaptive Robust Binary Pattern(ARBP)

Adaptive Robust Binary Pattern(ARBP) (X. Wu et al. 2018)[44] follows the similar concepts as LBP & AMBP. It contains three steps - determination of the optimal size of the analysis window, determination of the local threshold value

Algorithm 1: Adaptive Median Binary Patterns(AMBP)

Input : Gray scale Image$I$; maximum analysis window k_{max}; patch size$L = 9$;
Output: Adaptive Median Binary Pattern Image, M;
for ***all*** i, j **do**
    $k \leftarrow 1$;
    repeat
        $S \leftarrow I[i-k:i+k, j-k:j+k]$;
        $Z_{med} \leftarrow$ median(S) ;
        $Z_{min} \leftarrow$ min(S);
        $Z_{max} \leftarrow$ max(S);
        if $Z_{min} < Z_{med} < Z_{max}$ **then**
            break
        end if
        $k \leftarrow k + 1$;
    until $k \leqslant k_{max}$;
    if $Z_{min} < I[i, j] < Z_{max}$ **then**
        $\tau = I[i, j]$
    end if
    else
        $\tau = Z_{med}$
    end if
    $M[i, j] \leftarrow L-$ bit binary pattern using τ and Equation 2.19
end for

and encoding.

In contrast to AMBP, here a new parameter is introduced Z_{para} that is used to generate the threshold τ. This parameter can be selected from Z_{ave} and Z_{med} whichever has the furthest distance from the center pixel. Also, here Z_{ave} defines the average intensity. The equation of ARBP is given here

$$ARBP(P,R) = \sum_{p=0}^{P-1} 2^k s(g_p - \tau) \qquad s(x) = \begin{cases} 1, & x \geqslant 0 \\ 0, & \text{Otherwise} \end{cases} \tag{2.20}$$

Here the value of s, g_p and x is already defined in the equation 2.19 but threshold τ is calculated differently in the Algorithm 2. The algorithm for ARBP is given down here which shows the step by step guide for the descriptor.

Algorithm 2: Adaptive Robust Binary Pattern(ARBP)

Input : Gray scale Image I; maximum analysis window k_{max}; patch size$L = 9$;

Output: Adaptive Robust Binary Pattern Image, M;

for ***all*** i, j **do**

  $k \leftarrow 1$;

  repeat

    $S \leftarrow I[i-k : i+k, j-k : j+k]$;

    $Z_{med} \leftarrow$ median(S) ;

    $Z_{min} \leftarrow$ min(S);

    $Z_{max} \leftarrow$ max(S);

    $Z_{ave} \leftarrow$ average(S);

    $d = \frac{1}{|S|} \sum_{m \epsilon S(i,j)} x_c - x_m, m \neq c$;

    if $|Z_{med} - I[i,j]| < |Z_{ave} - I[i,j]|$ **then**

      $Z_{para} = Z_{ave}$;

    end if

    else

      $Z_{para} = Z_{med}$;

    end if

    if $Z_{min} + \alpha.d < Z_{para} < Z_{max} - \alpha.d$ **then**

      break

    end if

    $k \leftarrow k + 1$;

  until $k \leqslant k_{max}$;

  if $Z_{min} + \alpha.d < I[i,j] < Z_{max} - \alpha.d$ **then**

    $\tau = I[i,j]$

  end if

  else

    $\tau = Z_{para}$(τ is the parmater used as the threshold)

  end if

  $M[i,j] \leftarrow L -$ bit binary pattern

end for

Limitations

It does better job than AMBP but still noise can affect its performance.

2.5.3 Directional Age Primitive Pattern(DAPP)

DAPP(M. T. B. Iqbal et al., 2017)[45] efficiently represents both cranio-facial growth(shape) & skin aging (texture) information. Initially 3 states are considered- (1) is set for Wrinkle textures, (2) is set for lip/eye corners, otherwise it is set to 0 as:

$$P_{init}(x,y) = \begin{cases} 1, & \text{if } \theta = \tau_1 \\ 2, & \text{if } \theta = \tau_2 \\ 0, & \text{otherwise} \end{cases} \tag{2.21}$$

Here, τ_1 refers to 45° and τ_2 refers to 90° as shown in figure 2.10.

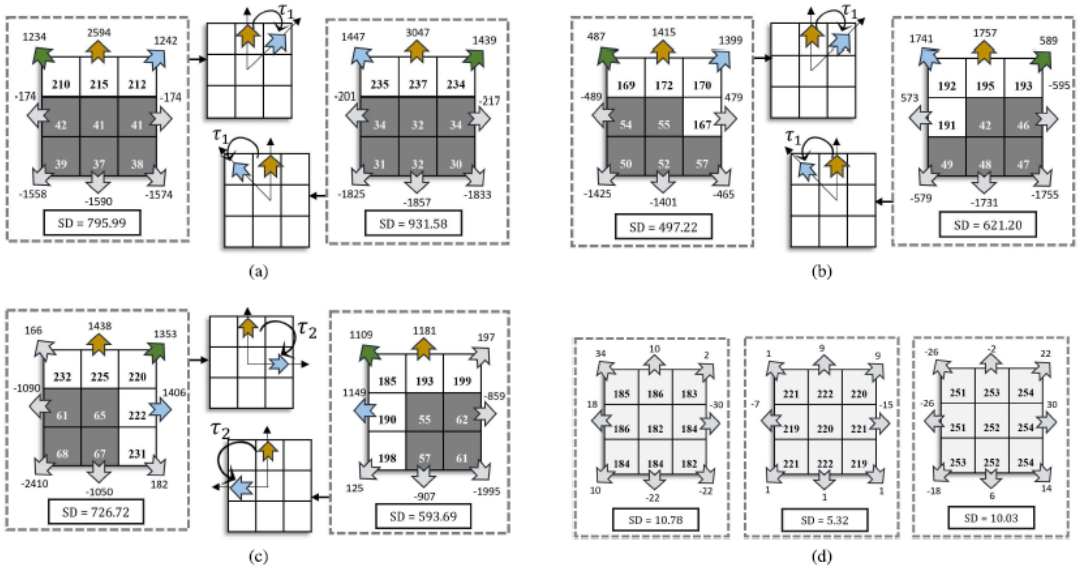

Figure 2.10: (a). Straight wrinkle (b) Curved Wrinkle (c) Eye corner (d) Flat patch

Wrinkles can appear either as straight edges as in 2.10(a) or curved edges

as shown in figure 2.10 (b). Moreover, secondary and tertiary edge response can be clockwise or counter clockwise to each-other. All these information extends these three categories into six as equation .

$$s(x) = \begin{cases} 1, & if,\ Diff_{ST} < \gamma \\ 2, & if,\ Diff_{ST} \geqslant \gamma,\ dir_1 \circlearrowleft dir_2 \\ 3, & if,\ Diff_{ST} \geqslant \gamma,\ dir_1 \circlearrowright dir_2 \\ 4, & if,\ dir_1 \circlearrowleft dir_2 \\ 5, & if,\ dir_1 \circlearrowright dir_2 \\ 0, & if,\ P_{init} = 0 \end{cases} \tag{2.22}$$

Threshold is generated as the highest accumulated bin of histogram of Standard Deviation (SD) of edge responses in cheek region. DAPP code generation is shown in figure 2.11

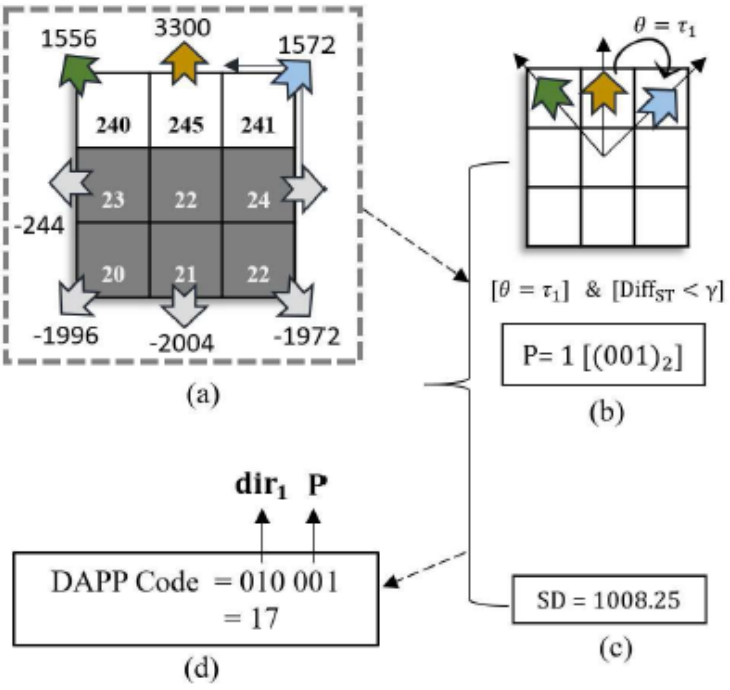

Figure 2.11: (a). Sample image patch and its Kirsch responses. (b) Age-Primitive code computation (c) Adaptive Thresholding (d) Final DAPP computation

Here in figure 2.11, Kirsch masks are applied to get eight directional edge response. Then top three edge responses are selected as primary, secondary and tertiary edge response. Then final age-primitive code, P is generated using equation 2.22. Here angle between primary and secondary edge response is 45°, so θ is τ_1 here. Difference between secondary and tertiary edge response is assumed to be less than the threshold σ. Therefore, final age-primitive code, P is 1 for this example. Then the direction of primary edge response and final age-primitive code is concatenated to generate final DAPP code that is 17 in this case.

Chapter 3

Proposed Feature Descriptor: Adaptive Robust Local Complete Pattern (ARLCP)

3.1 Overview

The recent challenge in facial expression recognition systems is to extract facial features which are robust in changing environment. In this chapter, we explain the proposed Adaptive Robust Local Complete Pattern (ARLCP), a new local pattern for effective facial feature representation, which includes the description of the basic ARLCP operator and how to construct facial feature descriptor based on ARLCP.

In our approach, we take images from datasets and then divide the images into $n \times n$ (e.g. 5×5, 10×10) regions. For each of the regions, we extract features and then create feature vector. Later on, we concatenate each of the vectors to generate final feature vector that represents an image.

3.2 Adaptive Robust Local Complete Pattern (ARLCP) Code

LBP, LTP and LBC methods use merely the gray level intensity values to encode the local texture of an image, which makes these methods unstable under the presence of large illumination variations. LBP is also susceptable to noise. Therefore, some recent local pattern operators exploit more stable gradient information or edge response values instead of gray levels. One such method is the Sobel-LBP that combines the LBP with the Sobel operator in order to enhance the local features and thus facilitates the extraction of more detailed information. Another approach is the local directinal pattern (LDP) [6] that employs eight directional edge response values to encode the local texture. However, since both Sobel-LBP and LDP methods use two discrimination levels (0 and 1) to generate binary texture patterns, these methods tend to produce inconsistent codes in uniform and smooth regions, where the intensity

variations among the neighbors are negligible. Also LDP dose not consider the magnitude of edge response. Another approach is the idea of completeness in micro-pattern considering the sign and magnitude of local difference along with the center intensity value. Such type of methods like CLBP, CLBC and CLTP still use the gray level intensity values which are susceptible to noise. Considering these limitations, we present the Adaptive Robust Local Positional Pattern (ARLCP) - a new local texture pattern for representing facial features. Instead of quantizing the gray level intensities like CLBP or CLTP, the ARLCP operator encodes the more stable edge esponse values of a local neighborhood, which retains more information of the local image content. Additionally, the proposed method uses adaptive threshold instead of a static one for each image as static threshold on primary response is quite sensitive since it may wipe-out either edge pixels in low-contrast image or flat pixels in natural noisy images

3.3 The Basic ARLCP Encoding Scheme

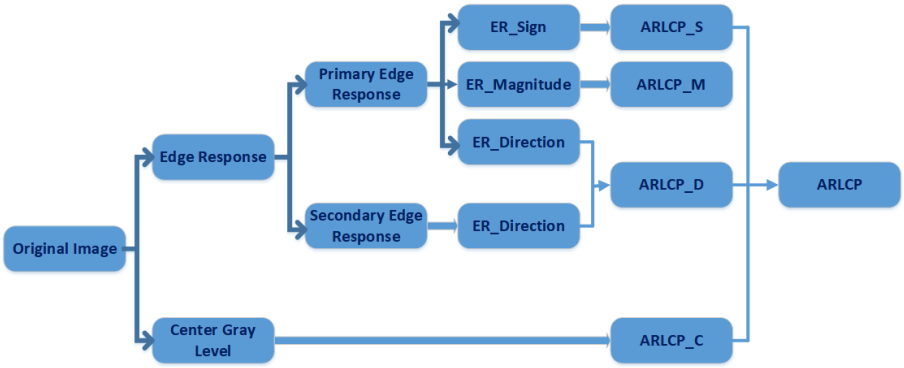

Figure 3.1: ARLCP Framework

Here we propose a face descriptor, Adaptive Robust Local Complete Pattern (ARLCP) as shown in figure 3.1, for generating feature representation of facial images which is robust in presence of noise and illumination variation. In the preprocessing step an image is first cropped based on the position of the eyes, nose and mouth and the face descriptor is applied on the resulting cropped image to generate a face representation. In our proposed method, we follow four steps to compute ARLCP feature vector:

i. Edge response computation

ii. Primary and secondary edge response selection

iii. Threshold(σ) generation

iv. Sign, Magnitude, Direction and Center codes generation

v. Histograms of Sign, Magnitude, Direction and Center codes generation

3.3.1 Step 1: Edge Response Computation

To get direction information of each pixel, we extract edge response values in eight different directions. We apply Kirsch compass mask shown in figure 3.2 for this purpose since it detects different directional edge responses more accurately than the others due to the consideration of all eight neighbors. We convolve the Kirsch mask in eight different orientations by rotating 45° where, I(x, y) is the pixel value, K_i is the i^{th} Kirsh mask among eight (K_0, K_1, ...K_7), as shown in 3.2. ER_i is their corresponding edge response values as shown in figure 3.3.

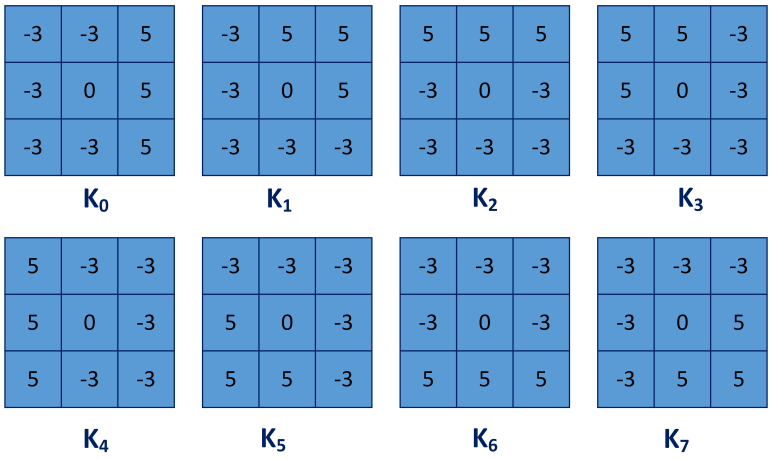

Figure 3.2: Kirsch Mask

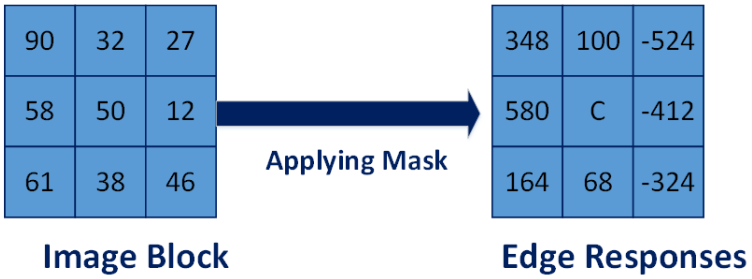

Figure 3.3: Edge Responses

3.3.2 Step 2: Primary and secondary edge response selection

Applying the 8 masks generates 8 responses each representing the edge significance in its respective direction. Among these eight edge responses, we consider two top most edge responses as they contain most significant edge information. Here edge response with highest and second highest values are considered as primary and primitive secondary edge response respectively. To convert primitive secondary edge response into secondary edge response, we consider the following two things.

a. if primitive secondary edge response is a neighbor to primary edge response, next highest edge response to primitive secondary edge response is considered as secondary edge response.

b. if primitive secondary edge response is not a neighbor to primary edge response, primitive secondary edge response itself is considered as secondary edge response.

Figure 3.4 illustrates the selection of primary and secondary edge responses.

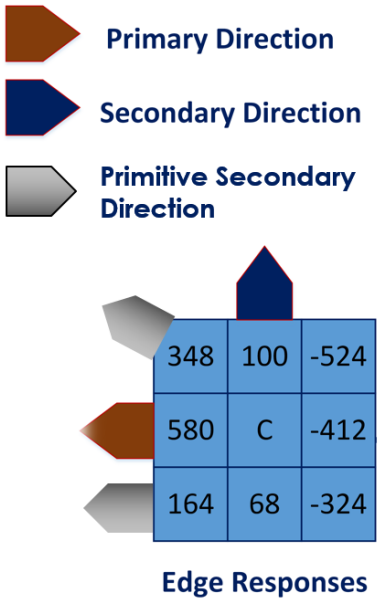

Figure 3.4: Primary and Secondary Edge Response Selection

3.3.3 Step 3: Threshold(σ) generation

We need a threshold to generate magnitude code as in equation . We introduce a global adaptive thresholding schema to differentiate the flat region from high-textured region.We carefully observe the characteristics of flat textures in a face image and found that, for a flat patch, edge responses are low and close to each other. It is also evident that, flat textures are dominant among all texture patterns in a face image and mostly appear in the cheek region. To generate threshold (σ), we take the median of the primary edge responses in the cheek region.

To find the cheek region as shown in figure 3.5, we use Active Appearance Model (AAM) that provides us 68 landmarks for each image and we select following four landmarks among them.

- most left eyebrow point
- most right eyebrow point
- most lower eye point

- most upper lip point

  These four landmarks form a rectangle that defines the cheek region.

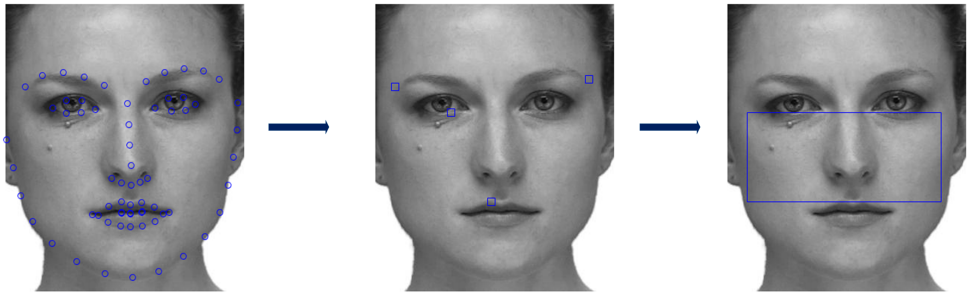

Figure 3.5: Cheek Region

3.3.4 Step 4: Sign, Magnitude, Direction and Center codes generation

Given a pixel, ARLCP Sign (ARLCP_S) is generated based on the information of sign of edge responses as:

$$ARLCP\_S = \sum_{i=0}^{p-1} s(E_i)2^i; \qquad s(x) = \begin{cases} 1, & \text{if } x \geqslant 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

Here, E_i refers to i^{th} edge response. If the edge response is positive, it is set to 1 and 0 otherwise. Then the binary pattern is encoded to decimal shown in figure 3.6 that represents the sign code of the pixel.

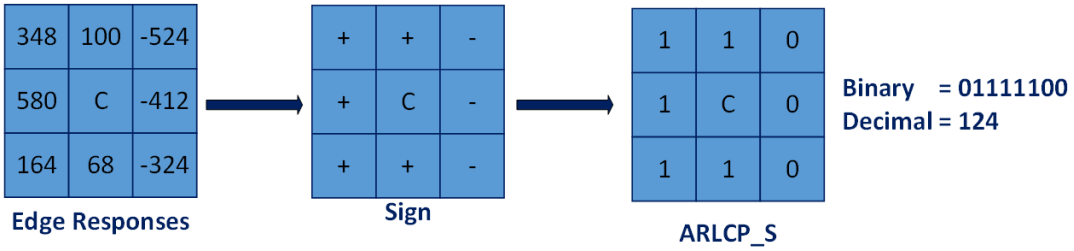

Figure 3.6: ARLCP_S generation

To get the information how significant the edge response in each direction, ARLCP Magnitude (ARLCP_M) code is generated that can be defined as:

$$ARLCP\_M = \sum_{i=0}^{p-1} s(|E_i| - \sigma)2^i \tag{3.2}$$

Here, σ is the adaptive threshold generated in step 3. Edge response with higher magnitude with respect to σ is considered as 1 and 0 otherwise. Figure 3.7 illustrates ARLCP Magnitude (ARLCP_M) code generation. Here, threshold σ has value of 196.

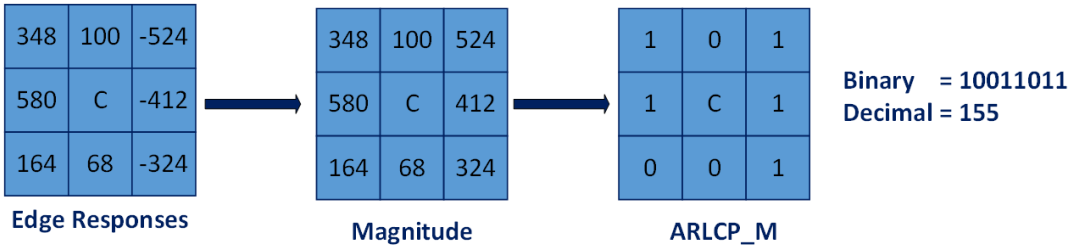

Figure 3.7: ARLCP_M generation

Along with the sign and magnitude of edge response, we also incorporate direction information. Figure 3.8(a) shows all the eight directions we have considered. Primary and secondary edge direction selected in step 2 are concatenated as equation (3.3) and generate ARLCP Direction (ARLCP_D) code as shown in figure 3.8(b) that represents the Direction code of the pixel.

$$ARLCP\_D = [d_1, d_2] \tag{3.3}$$

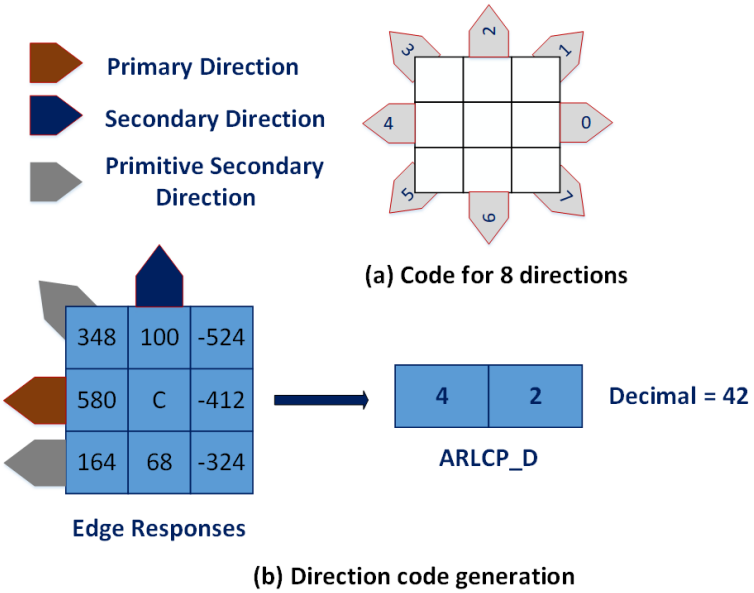

Figure 3.8: ARLCP$_D$ generation

Finally ARLCP Center (ARLCP_C) code is generated thresholding the given pixel with respect to average intensity value of entire image, (γ) shown in figure 3.9. Here, threshold γ has value of 64. ARLCP_C can be defied as:

$$ARLCP\_C = S(g_c - \gamma) \tag{3.4}$$

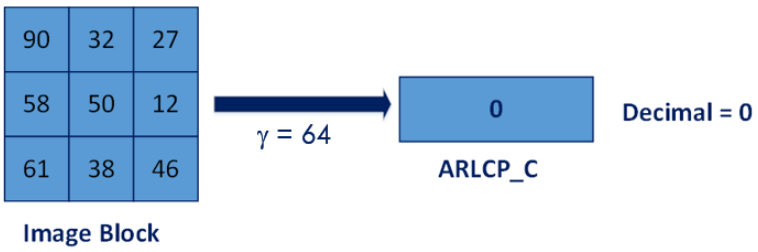

Figure 3.9: ARLCP_C generation

In essence, Figure 3.10 illustrates how all these four codes are generated from an 3x3 image block. The initial value in the block is the intensity of the

9 pixels in the block. Likewise each pixel of the image generates 8 directional responses. Edge response magnitudes are more stable than intensity values in presence of noise and illumination variation and the use of edge response in our method makes it more robust in presence of noise and lighting variation. After calculating the edge responses, we select top two edge responses defined as primary and primitive secondary edge response respectively. And we select secondary edge response avoiding neighbors of primary edge response. Then Sign and magnitude of edge responses are generated. Here ARLCP Sign (ARLCP_S) code is generated by converting negative values to zeros and positive values to ones. In case of ARLCP Magnitude (ARLCP_M) code generation, magnitudes greater or equal to threshold σ are considered as 1 and 0 otherwise. Here, σ is calculated from the median primary edge response in the cheek region. After that, we concatenate the directions of primary and secondary edge responses to get ARLCP Direction (ARLCP_D) code. Lastly we generate ARLCP Center (ARLCP_C) code by thresholding center pixel in a block of 3x3 neighborhood.

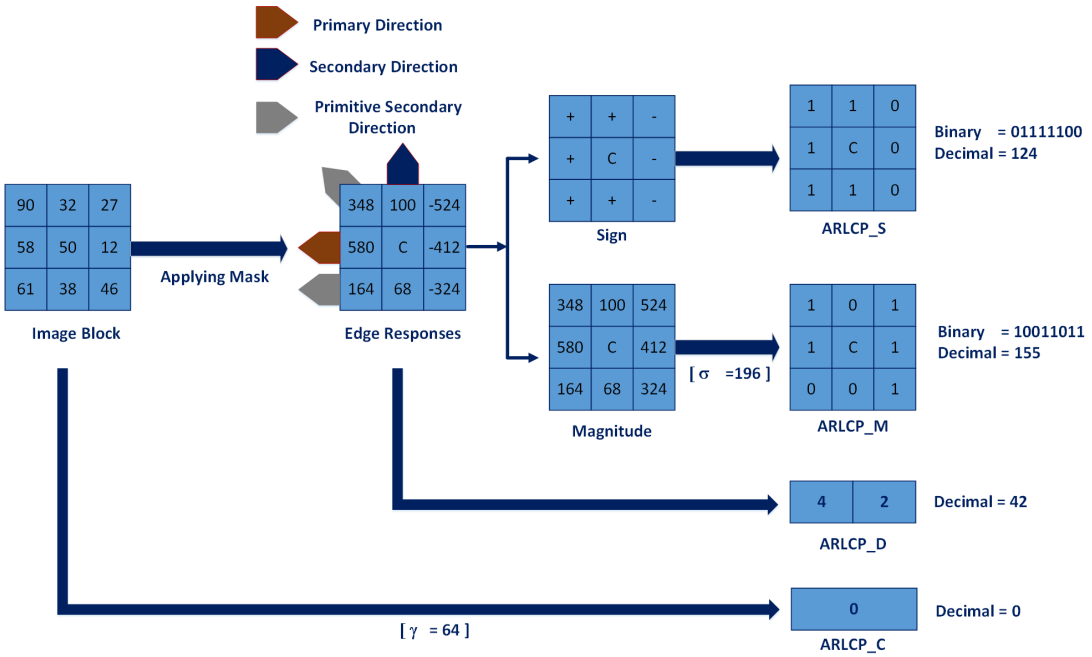

Figure 3.10: ARLCP codes generation

3.3.5 Step 5: Histograms of Sign, Magnitude, Direction and Center codes generation

Histogram is generated for each of the ARLCP_S, ARLCP_M, ARLCP_D, ARLCP_C code following the equation 3.5, 3.6 and concatenated to generate the final feature vector that represents an image. Figure 3.11 depicts the final feature generation. Here H^k represents histogram of k^{th} region.

$$H^k(c) = \sum_{(x,y)\in R^k} \partial(P(x,y)), c), \forall c \tag{3.5}$$

$$\partial\,(a,b) = \begin{cases} 1 & a = b \\ 0 & a \neq b \end{cases} \tag{3.6}$$

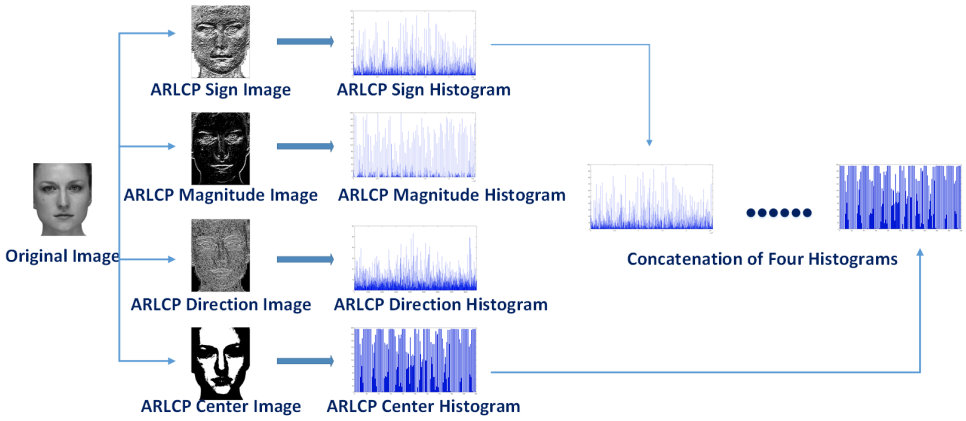

Figure 3.11: Histogram Generation

3.4 Facial Feature Representation with ARLCP

To retain more spatial information, an image is divided into $n \times n$ regions (e.g. 10×10, 5×5). Applying the ARLCP operator on all the pixels of a region will result in four ARLCP histograms as ARLCP_S, ARLCP_M, ARLCP_D and ARLCP_C. Then these four histograms are concatenated to form a feature vector of that region. Finally feature vectors for all regions are concatenated to generate final feature vector as shown in figure 3.12 that represents the image.

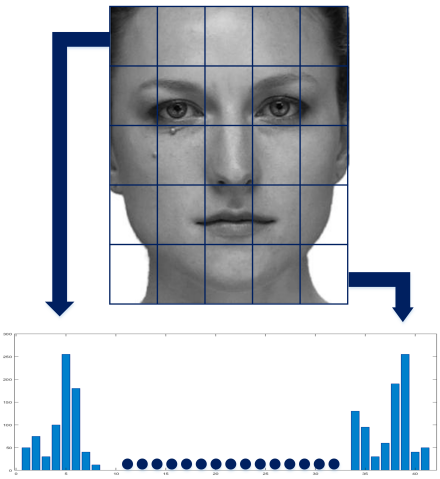

Figure 3.12: Local Histograms Concatenation

3.5 Expression Recognition using KNN Classifier

We experiment our proposed feature descriptor Adaptive Robust Local Complete Pattern (ARLCP) with K-Nearest Neighbor (KNN) to recognize different facial expressions. We take images and generate feature vector for each image using ARLCP and then use the feature vectors as input to KNN for classifying the class of the images.

The KNN algorithm is a robust and versatile classifier that falls in the supervised learning family of algorithms. Informally, this means that we are given a labelled dataset consiting of training observations (x,y) and would like to capture the relationship between x and y. More formally, our goal is to learn a function $h{:}X \rightarrow Y$ so that given an unseen observation $\boldsymbol{x}$, $\boldsymbol{h(x)}$ can confidently predict the corresponding output y.

When KNN is used for classification, the output can be calculated as the class with the highest frequency from the K-most similar instances. Each instance in essence votes for their class and the class with the most votes is taken as the prediction. Similarity is defined according to a distance metric between two data points. A popular choice is the Euclidean distance given by

$$d(x, x') = \sqrt{(x_1 - x'_1)^2 + (x_2 - x'_2)^2 + ... + (x_n - x'_n)^2} \tag{3.7}$$

Also there are other measures like Manhattan, Chebyshev and Hamming distance. More formally, given a positive integer K, an unseen observation x and a similarity metric d, KNN classifier performs the following two steps:

- KNN runs through the whole dataset computing d between x and each training observation. Lets say, A be the set of k points in the training data that are closest to x. K is usually odd to prevent tie situations.
- Then the conditional probability for each class is estimated, that is, the fraction of points in A with that given class label as:

$$P(y = j, X = x) = \frac{1}{K} \sum_{i \epsilon A} I(y^i = j) \tag{3.8}$$

Here, I(x) is the indicator function which evaluates to 1 when the argument x is true and 0 otherwise. The class with highest probability is assigned to the given sample.

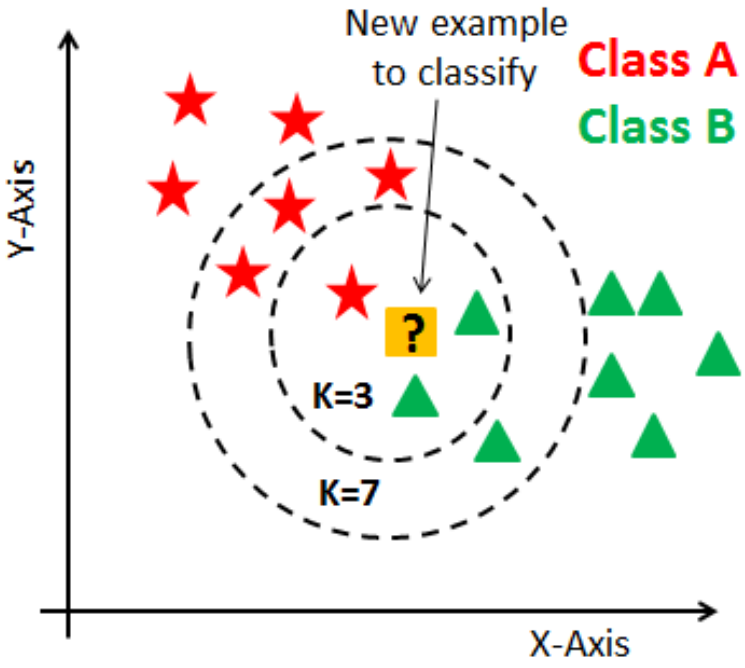

Figure 3.13: KNN Classification

Figure 3.13 illustrates how KNN classifier classifies a given sample with 2 dimensional feature for different values of K. Here, KNN classifies a given sam-

ple to class B when K=3 as within the three closest training samples, two samples are from class B. But in case of K=7, four out of seven training samples are from class A. So KNN classifies the given sample as class A. Therefore, classification can be different for different value of K.

3.6 Strengths of the Proposed Descriptor

Here we discuss the good properties of our proposed face descriptor for facial expression recognition. Some strengths of the proposed feature descriptor are:

- Instead of direct pixel, this method uses directional edge response that is more robust to noise and illumination variation.
- This method uses directional information along with sign and magnitude of edge responses that leads to more unique representation of an image.
- Instead of static threshold for all image, this method uses an adaptive threshold for each image as static threshold on primary response is quite sensitive since it may wipe-out either edge pixels in low-contrast image or flat pixels in natural noisy images

Chapter 4

Experimental Result

4.1 Experimental Setup and Dataset Description

The JAFFE database [46] comprises facial expression images of 10 Japanese female subjects. All the images were digitized into a resolution of 256x256 pixels. The images were taken from a frontal pose, and the subjects' hair was tied back in order to facilitate the exposure of all the expressive zones of the face. In the image scene, an even illumination was created using tungsten lights. Instead of revealing the actual names, the subjects are referred with their initials, which are KA, KL, KM, KR, MK, NA, NM, TM, UY, and YM. In our setup,the expression dataset comprises a total of 213 images.

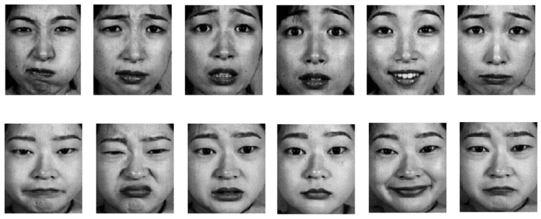

Figure 4.1: JAFFE Database

The Extended Cohn-Kanade Facial Expression (CK+) database [47] comprises 593 image sequences (from neutral to apex) of 123 subjects who were instructed to perform a series of 23 facial displays. From these sequences, 327 out of 593 have each of the seven basic emotion categories: anger, contempt, disgust, fear, happiness, sadness and surprise. In our setup, we selected 327 sequences with 7 emotion categories.The three most expressive image frames were selected from each sequence to make the 7-class expression dataset (981 images).

Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) We use 4904 videos from RAVDESS dataset [48]. We collect 50 frames from each videos that leads to total 245,200 images with 8 emotion categories (neu-

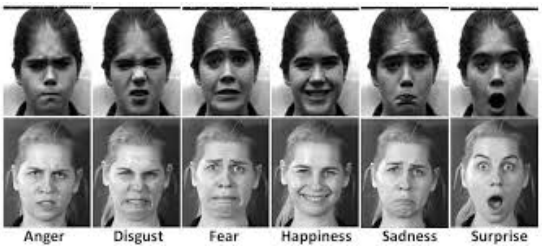

Figure 4.2: CK+ Database

tral, calm, happy, sad, angry, fearful, disgust, surprised). The selected images were cropped from the original and then normalized to 320x380 pixels.

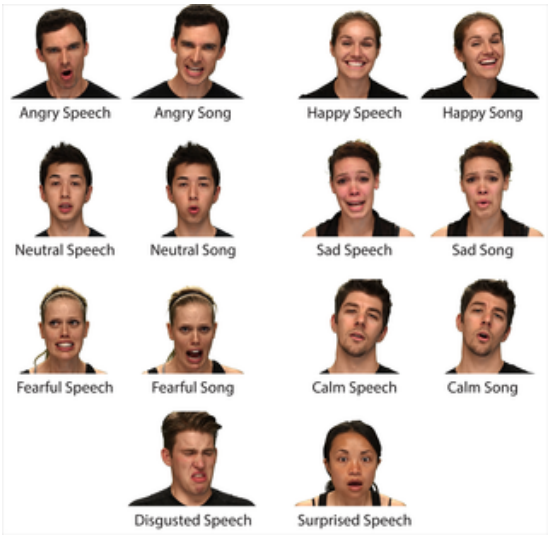

Figure 4.3: Ravdess Database

4.2 Performance Evaluation

In order to evaluate the proposed method in expression recognition, we carried out a five-fold cross-validation to measure the classification rate. In a five-fold cross-validation, the whole dataset is randomly partitioned into five subsets, where each subset comprises an equal number of instances. One subset is used as the testing set and the classifier is trained on the remaining nine subsets. The average classification rate is calculated after repeating the above process for five times. K-Nearest Neighbor is used as the classifier. We present the performance of the experiments with accuracy which is defined as:

$$\text{Accuracy} = \frac{\text{Number of correctly classified samples}}{\text{Total number of test samples}}$$

4.3 Performance on JAFFE Database

For the JAFFE database, we have compared the performance of the proposed method with other well-known local pattern operators, namely local binary pattern (LBP) [49], local ternary pattern (LTP) [50], local directional pattern (LDP) [51], Complete Local Binary Pattern(CLBP)[39] and Adaptive Robust Binary Pattern([44]) . We used K Nearest Neighbor (KNN) to classify the test samples and measure the accuracy by counting the correctly classified samples. We used five-fold cross validation scheme to evaluate the performance.

Table 4.1: Recognition rate (%) for the JAFFE 7-class expression dataset.

Feature Descriptor	**Accuracy (%)**
LBP	88.21
LTP	88.67
LDP	89.52
CLBP	88.83
ARBP	92.51
ARLCP	94.41

Table 4.1 shows the recognition rate of our proposed ARCLP along with local pattern-based feature descriptors LBP, LTP, LDP, CLBP and ARBP against the 7-class expression JAFFE dataset. It can be observed that, ARLCP exhibits superior performance in recognizing expression images. However, JAFFE dataset contains fewer images than CK+ dataset. So we cannot ensure sufficient sample image for training and some expressions are labeled incorrectly or expressed wrongly. Therefore, the results from JAFFE dataset should be lower than CK+ dataset.

4.4 Performance on CK+ Database

For the CK+ database, the ARLCP feature descriptor achieves classification rate of 98.67%. Table 4.2 shows the recognition rates of different feature descriptors with the CK+ 7-class expression dataset. It can be observed that, ARLCP achieves the highest classification rate.

Table 4.2: Recognition rate (%) for the CK+ 7-class expression dataset.

Feature Descriptor	**Accuracy (%)**
LBP	90.21
LTP	95.67
LDP	95.73
CLBP	96.94
ARBP	97.56
ARLCP	98.67

We generate confusion matrix for CK+ dataset using ARLCP feature descriptor shown in table 4.3 where diagonal values define the matching with the original data and non-diagonal values show the mismatch.

Table 4.3: Confusion matrix for the CK+ 7 dataset -class recognition using ARLCP feature representation.

		1	2	3	4	5	6	7
1	Anger	**96.4**	0	3.6	0	0	0	0
2	Contempt	0	**100**	0	0	0	0	0
3	Disgust	0	0	**97.34**	0	2.66	0	0
4	Fear	0	0	0	**100**	0	0	0
5	Happy	0	0	0	0	**100**	0	0
6	Sadness	0	0	3.05	0	0	**96.95**	0
7	Surprise	0	0	0	0	0	0	**100**

Here we can see here, the expression "Anger" has 96.4% accuracy and 3.6% of it mismatches with the disgust expressions as they are very close. Similarly, 2.66% of "Disgust" expressions have been matched with the "Happy" faces. Also, "Sadness" conflicted with the "Disgust" faces at 3.05%.

To investigate the robustness of the proposed ARLCP descriptor under the presence of noise, further experiments are conducted on the images from the CK+ 7-class expression dataset. In the experimental setup, the images are contaminated with Gaussian noise of different variances. Figure 4.4 shows the recognition rate for images corrupted with noise of mean 0 and different variances (0.02, 0.03 and 0.04), experimented with different operators. We have observed that ARLCP provides better accuracy than other existing local oper-

ators.

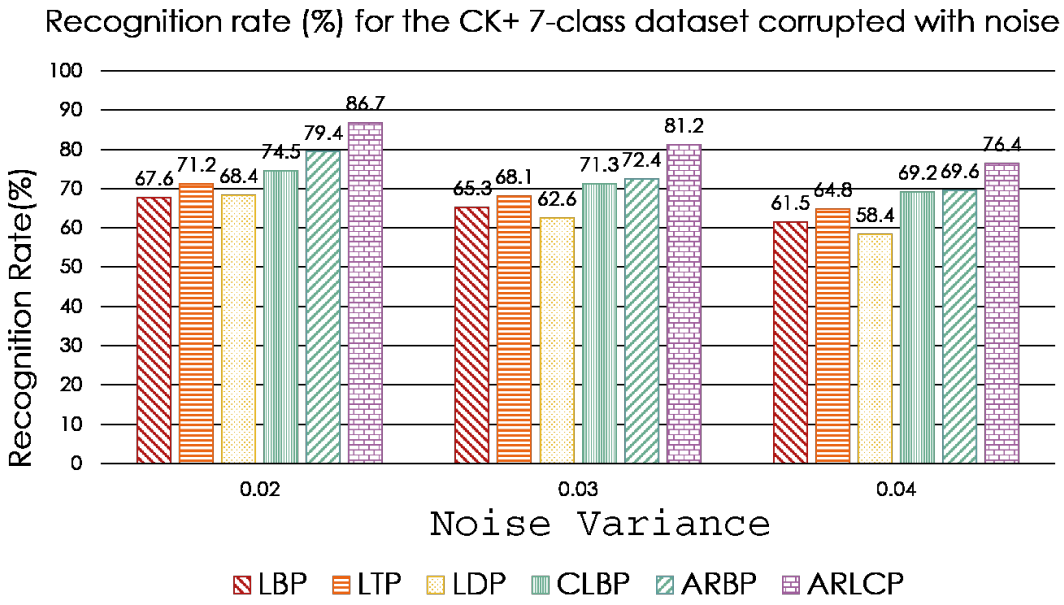

Figure 4.4: Recognition rate (%) for the CK+ 7-class dataset corrupted with noise

Our proposed method ARLCP also shows better performance in case of lower resolution. Automated facial analysis is useful in smart meeting, surveillance and many other applications, where often only low-resolution video data is available. Since geometric methods like detection of facial action units are difficult to accommodate in these scenarios, appearance-based methods seem to be a better solution. Therefore, the performance of the proposed method is also evaluated on low-resolution images. We experiment with different resolution and observe that our method exhibits superior performance. Figure 4.5 shows the recognition rate for CK+ dataset with different resolution.

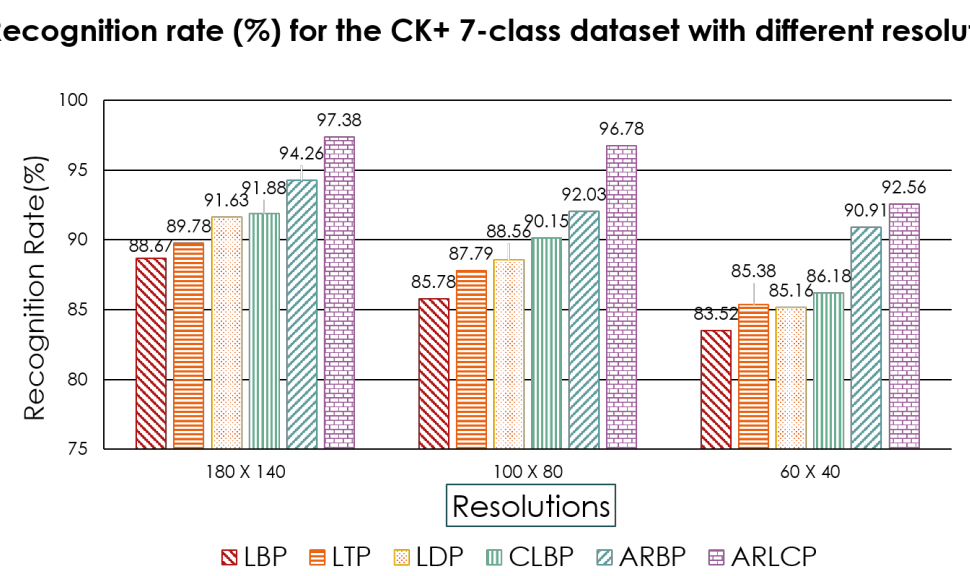

Figure 4.5: Recognition rate (%) for the CK+ 7-class dataset with different resolutions

It is important to retain spatial information as much as possible and it can be achieved by dividing the images into more number of regions. Hence, we also experimented our proposed method dividing the images with different number of regions such as 10×10, 5×5 and 2×2. However, excessively increasing the number of regions may result in the decrease of accuracy as there will be less number of pixels in each region. Here we compare ARLCP with other methods and it can be observed that ARLCP provides better accuracy. Figure 4.6 shows the recognition rate for CK+ dataset divided by several number of regions.

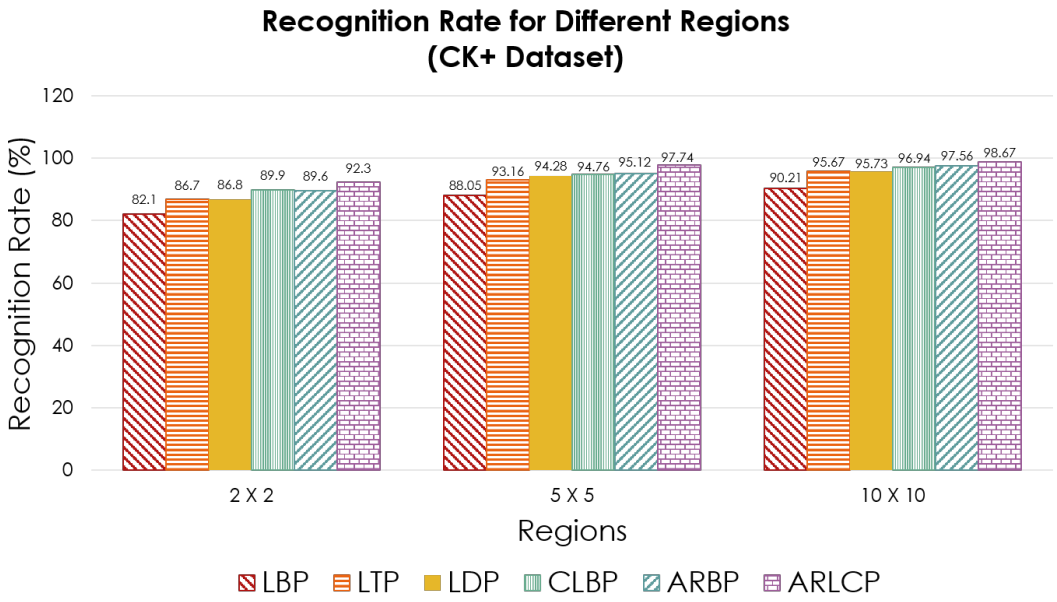

Figure 4.6: Recognition Rate(%) for Different Regions (CK+ Dataset)

4.5 Performance on RAVDESS Database

For the RAVDESS dataset, ARLCP achieves Recognition rate of 99.90% and 99.86% using KNN classifier with 10-Fold and 5-fold cross validation respectively. Table 4.4 shows the confusion matrix for RAVDESS dataset with 5-fold cross validation.

Table 4.4: Confusion matrix for the RAVDESS 8-class recognition using ARLCP feature representation with 10-fold cross validation

		1	2	3	4	5	6	7	8
1	Neutral	**99.8**	0.13	0	0	0.07	0	0	0
2	Calm	0.08	**99.8**	0.03	0	0.03	0	0.03	0.03
3	Happy	0	0	**100**	0	0	0	0	0
4	Sad	0	0	0	**99.9**	0	0.1	0	0
5	Angry	0	0	0	0	**99.9**	0.05	0.05	0
6	Fearful	0.03	0	0.03	0	0.14	**99.8**	0.03	0
7	Disgust	0	0	0	0.05	0.05	0	**99.9**	0
8	Surprised	0	0	0	0	0	0	0	**100**

Chapter 5

Conclusion

5.1 Research Summary

The thesis aims to design a robust facial feature representation that overcomes the limitations of existing approaches and can be used in facial expression recognition.

In this thesis we introduced a new local texture pattern, the Adaptive Robust Local Complete Pattern (ARLCP). The descriptor uses sign, magnitude and directional information of edge responses.Therefore, this operator is more robust to noise and illumination variation. The ARLCP operator also integrates adaptive threshold instead of a static one as static threshold on primary edge response is quite sensitive to noisy and low-contrast images. The proposed method has been evaluated for facial expression recognition classification. Experiments using different benchmark datasets exhibits better performance of ARLCP against other well-known appearance-based feature descriptors namely LBP, LTP, LDP, CLBP, ARLBP. As four histograms are concatenated to generate final feature vector, execution time increases.

The ARLCP operator provides robustness to noise and illumination variation, higher discriminating capability compared to existing feature descriptors. Therefore, it can be effectively used in facial expressions recognition.

5.2 Future Works

In our proposed method, we considered static images for facial expression recognition. In future, we plan to use our method for facial expression recognition from video sequence images. Also we plan to optimize the method to reduce the feature dimension as well as time complexity.

References

[1] "Digaital Camera -wikipedia," https://en.wikipedia.org/wiki/Digital_camera, accessed: November 14, 2018.

[2] "How image processing will change your world in future -the economic times," https://economictimes.indiatimes.com/tech/software/how-image-processing-will-change-your-world-in-future/articleshow/10394958.cms, accessed: November 14, 2018.

[3] S. R. Benedict and J. S. Kumar, "Geometric shaped facial feature extraction for face recognition," *2016 IEEE International Conference on Advances in Computer Applications (ICACA)*, 2016.

[4] "This App Uses Facial Recognition Software to Help Identify Genetic Conditions -," https://www.smithsonianmag.com/innovation/app-uses-facial-recognition-software-help-identify-genetic-conditions-180961897/, accessed: November 14, 2018.

[5] T. Jabid, M. H. Kabir, and O. Chae, "Local directional pattern (ldp) for face recognition," in *2010 Digest of Technical Papers International Conference on Consumer Electronics (ICCE)*, Jan 2010, pp. 329–330.

[6] Wikipedia contributors, "Geometric feature learning — Wikipedia, the free encyclopedia," 2018, [Online; accessed 15-October-2018]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Geometric_feature_learning&oldid=845563063

[7] P. Ekman and W. Friesen, "Facial action coding system: A technique for measurement of facial movement," 1978.

[8] Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu, "Feature-based facial expression recognition: Sensitivity analysis and experiments with a multilayer perceptron (vol 13, pg 893, 1999)," vol. 14, pp. 257–257, 03 2000.

[9] G. Guo and C. R. Dyer, "Simultaneous feature selection and classifier training via linear programming: a case study for face expression recog-

nition," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, vol. 1, June 2003, pp. I–I.

[10] M. F. Valstar, I. Patras, and M. Pantic, "Facial action unit detection using probabilistic actively learned support vector machines on tracked facial point data," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, Sept 2005, pp. 76–76.

[11] L. Wiskott, N. Krüger, N. Kuiger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775–779, July 1997.

[12] B. Poggio, R. Brunelli, and T. Poggio, "Hyberbf networks for gender classification."

[13] B. E. H. Abdi, D. Valentin and A. O'Toole, "More about the difference between men and women: Evidence from linear neural network and principal component approaches," *Pattern Recognition*, vol. 7, no. 6, pp. 1160–1164, 1995.

[14] B. E. H. Abdi, D. Valentin and A. O'Toole, "Representation face images for emotion classification," *Advances in Neural Information Processing Systems*, vol. 9, pp. 894–900, 1997.

[15] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski, "Face recognition by independent component analysis," *IEEE Transactions on Neural Networks*, vol. 13, no. 6, pp. 1450–1464, Nov 2002.

[16] C. C. Fa and F. Y. Shin, "Recognizing facial action units using independent component analysis and support vector machine," *Pattern Recognition*, vol. 39, no. 9, pp. 1795–1798, 2006.

[17] Y. li Tian, "Evaluation of face resolution for expression analysis," in *2004 Conference on Computer Vision and Pattern Recognition Workshop*, June 2004, pp. 82–82.

[18] M. J. Lyons, J. Budynek, and S. Akamatsu, "Automatic classification of single facial images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 12, pp. 1357–1362, Dec 1999.

[19] M. Z. Uddin, J. J. Lee, and T. . Kim, "An enhanced independent component-based human facial expression recognition from video," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 4, pp. 2216–2224, November 2009.

[20] A. H. T. Ahonen and M. Pietikainen, "Face recognition with local binary pattern," *European Conference on Computer Vision*, 2004.

[21] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 1991, pp. 586–591.

[22] K. Etemad and R. Chellappa, "Dicriminant analysis for recognition of human face images," *Journal of the Optical Society of America*, vol. 14, 1997.

[23] J. Yang, D. Zhang, A. F. Frangi, and J. yu Yang, "Two-dimensional pca: a new approach to appearance-based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131–137, Jan 2004.

[24] P. Penev and J. Atick, "Local feature analysis: A general statistical theory for object representation," *Network: Computation in Neural Systems*, vol. 7, 1996.

[25] M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Wurtz, and W. Konen, "Distortion invariant object recognition in the dynamic link architecture," *IEEE Transactions on Computers*, vol. 42, no. 3, pp. 300–311, March 1993.

[26] S. Zhao, Y. Gao, and B. Zhang, "Sobel-lbp," in *2008 15th IEEE International Conference on Image Processing*, Oct 2008, pp. 2144–2147.

[27] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, "Classifying facial actions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 974–989, Oct 1999.

[28] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, Dec 2006.

[29] D. Huang, C. Shan, M. Ardabilian, and L. Chen, "Facial image analysis based on local binary patterns: A survey," 10 2018.

[30] G. Zhao and M. Pietikäinen, "Boosted multi-resolution spatiotemporal descriptors for facial expression recognition," *Pattern Recognition Letters*, vol. 30, no. 12, pp. 1117 – 1127, 2009, image/video-based Pattern Analysis and HCI Applications. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865509000695

[31] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1635–1650, June 2010.

[32] M. H. K. T. Jabid and O. Chae, "Robust facial expression recognition based on local directional pattern,," vol. 32, pp. 784–794, 05 2010.

[33] M. P. T. Ojala and D. Harwoods, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29A, 1996.

[34] W. J. Yang Zhao, De-Shuang Huang, "Completed local binary count for rotation invariant texture classification," *IEEE TRANSACTIONS ON IMAGE PROCESSING*, vol. 24, 2012.

[35] M. P. T. Ojala and T. T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary pattern," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, 2002.

[36] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1635–1650, June 2010.

[37] B. Zhang, S. Shan, X. Chen, and W. Gao, "Histogram of gabor phase patterns (hgpp): a novel object representation approach for face recognition." *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 16, no. 1, pp. 57–68, Jan 2007.

[38] J. Ren, X. Jiang, and J. Yuan, "Relaxed local ternary pattern for face recognition," in *2013 IEEE International Conference on Image Processing*, Sept 2013, pp. 3680–3684.

[39] a. Z. Z.Guo, L. Zhang, "Acompletedmodeling of local binary pattern operator for texture classification," *IEEE TRANSACTIONS ON IMAGE PROCESSING*, vol. 24, 2010.

[40] M. Varma and A. Zisserman, "Texture classification: are filter banks necessary?" in *CVPR03*, 2003, pp. II: 691–698.

[41] Y. Zhao, D. Huang, and W. Jia, "Completed local binary count for rotation invariant texture classification," *IEEE Transactions on Image Processing*, vol. 21, no. 10, pp. 4492–4497, Oct 2012.

[42] T. Rassem and B. E. Khoo, "Completed local ternary pattern for rotation invariant texture classification," vol. Volume 2014 (2014), p. 10, 04 2014.

[43] A. Hafiane, K. Palaniappan, and G. Seetharaman, "Adaptive median binary patterns for texture classification," in *2014 22nd International Conference on Pattern Recognition*, Aug 2014, pp. 1138–1143.

[44] X. Wu, Q. Qiu, Z. Liu, Y. Zhao, B. Zhang, Y. Zhang, X. Wu, and J. Ren, "Hyphae detection in fungal keratitis images with adaptive robust binary pattern," *IEEE Access*, vol. 6, pp. 13 449–13 460, 2018.

[45] M. T. B. Iqbal, M. Shoyaib, B. Ryu, M. Abdullah-Al-Wadud, and O. Chae, "Directional age-primitive pattern (dapp) for human age group recog-

nition and age estimation," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2505–2517, Nov 2017.

[46] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets," in *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*. IEEE, 1998, pp. 200–205.

[47] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*. IEEE, 2010, pp. 94–101.

[48] S. R. Livingstone and F. A. Russo, "The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english," *PloS one*, vol. 13, no. 5, p. e0196391, 2018.

[49] C. Shan, S. Gong, and P. W. McOwan, "Robust facial expression recognition using local binary patterns," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 2. IEEE, 2005, pp. II–370.

[50] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE transactions on image processing*, vol. 19, no. 6, pp. 1635–1650, 2010.

[51] T. Jabid, M. H. Kabir, and O. Chae, "Robust facial expression recognition based on local directional pattern," *ETRI journal*, vol. 32, no. 5, pp. 784–794, 2010.