Image Analysis and Understanding

Haythem Ghazouani

République Tunisienne
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Carthage
École Nationale d'Ingénieurs de Carthage

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Habilitation Universitaire
in Computer Science

Image Analysis and Understanding

Application to texture classification, facial expression recognition and breast cancer diagnosis

by
Haythem Ghazouani

Assistant Professor at ENICarthage
Member of the LIMTIC Laboratory (ISI)

defended on February 1st, 2023, with the following committee members:

Afef Abdelkrim	Professor at ENICarthage	President
Imed Riadh Farah	Professor at ISAMM	Reviewer
Mohamed Ali Mahjoub	Professor at ENISo	Reviewer
Walid Barhoumi	Professor at ENICarthage	Examiner
Faten Chaieb	Associate Professor, Efrei Paris	Examiner

République Tunisienne
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Carthage
École Nationale d'Ingénieurs de Carthage

Synthèse des Travaux de Reccherche
Présentée en vue de l'obtention de

Habilitation Universitaire

Discipline : Informatique

Analyse et Compréhension de l'Image

Application à la classification de texture, à la reconnaissance des expressions faciales et au diagnostic du cancer du sein

---

Présenté par

Haythem Ghazouani

Maître Assistant à l'ENICarthage
Membre du laboratoire LIMTIC (ISI)

Soutenu le 01 Février 2023 devant le jury composé de :

Afef Abdelkrim	Professeure à l'ENICarthage	Présidente
Imed Riadh Farah	Professeur à l'ISAMM	Rapporteur
Mohamed Ali Mahjoub	Professeur à l'ENISo	Rapporteur
Walid Barhoumi	Professeur à l'ENICarthage	Examinateur
Faten Chaieb	Maître de Conférences, Efrei Paris	Examinatrice

Abstract

As one of the most active research areas in computer vision, image analysis and understanding attempts to detect low-level and high-level features, to locate, recognize objects, to detect anomalies and to classify them into classes and categories from images and videos. This dissertation focuses on the automation of visual feature extraction, selection and fusion for image classification. The main contribution presented in this manuscript is the fully automation of the process of local feature extraction and aggregation using genetic programming with different applications ranging from texture classification to breast cancer diagnosis including facial expression recognition. More precisely, low-level texture features are defined based on edge arrangements and automatically aggregated for texture image classification under different changes. The same framework is used to extract texture cues from human faces and fuse them with geometric features representing face landmark distances in order to capture wrinkles and face distortions to detect human affects. Facial expression recognition from 3D/4D facial images is also performed based on mesh-local binary pattern difference descriptor representing a unified set of geometric and appearance features of different facial regions. Texture is explored more intensely in breast tissue from mammography images to diagnosis cancer. A more powerful texture description is proposed to detect malignant tumor in breast tissue. A fully automated framework based on genetic programming for feature extraction, selection and fusion is also presented to perform content based retrieval and breast cancer diagnosis. For all the investigated applications, the presented frameworks perform training with small number of instances and tackle the problem of the unavailability of labeled data.

Résumé

L'analyse et la compréhension d'images est un domaine actif de la vision par ordinateur qui vise à détecter des caractéristiques de bas niveau et de haut niveau, à localiser et à reconnaître des objets, à détecter des anomalies et à les classer en classes et catégories à partir d'images et de vidéos. Cette dissertation porte sur l'automatisation de l'extraction, de la sélection et de la fusion de caractéristiques visuelles pour la classification d'images. La principale contribution présentée dans ce manuscrit est l'automatisation complète du processus d'extraction et d'agrégation de caractéristiques locales à l'aide de la programmation génétique. L'apport de cette contribution a été exploré dans diverses applications, allant de la classification de texture à la détection du cancer du sein en passant par la reconnaissance des expressions faciales. Plus précisément, des caractéristiques de texture de bas niveau sont définies à partir des dispositions des contours et sont par la suite agrégées de manière automatique pour la classification d'images de texture. Le même Framework est utilisé pour extraire des indices de texture locaux à partir de visages humains. Ces indices sont automatiquement fusionnés avec des caractéristiques géométriques, représentant les distances des points de repère du visage, pour capturer les rides et les distorsions du visage, afin de détecter les émotions humaines. La reconnaissance des expressions faciales à partir d'images faciales 3D/4D est également abordée en proposant un descripteur basé sur la différence de motif binaire local de maillage définissant un ensemble unifié de caractéristiques géométriques ainsi que d'apparence de différentes régions faciales. La texture est étudiée de manière plus approfondie dans les tissus mammaires des images de mammographie pour le diagnostic du cancer du sein. Une description de texture plus robuste est proposée pour détecter les tumeurs malignes dans les tissus mammaires. Un Framework entièrement automatisé basé sur la programmation génétique pour l'extraction, la sélection et la fusion de caractéristiques est également présenté pour réaliser une recherche par le contenu ainsi que le diagnostic de cancer du sein. Pour toutes les applications explorées dans ces travaux, les algorithmes présentés effectuent la phase d'entraînement avec un nombre réduit d'exemples pour aborder le problème de l'indisponibilité de données étiquetées.

Acknowledgements

I would like to express my heartfelt gratitude to all those who have supported me during the last years.

I am deeply grateful to Ms. Afef Abdelkrim, Professor at ENICarthage, for the honor of presiding over the jury of this university habilitation.

I would like to especially thank Mr. Imed Riadh Farah, Professor at ISAMM and Mr. Mohamed Ali Mahjoub, Professor at the ENISo for the great honor of accepting to report on this university habilitation.

I am deeply grateful to Ms. Faten Chaieb, Associate Professor at Efrei Paris, for accepting to be a part of the jury, and for her interest in my work.

I would also like to thank Mr. Walid Barhoumi, Professor at ENICarthage, for accepting to be a part of the jury, and for his continuous support, encouragement and confidence in my work.

I would also like to thank the members of LIMTIC laboratory, for their valuable insights and suggestions, which have greatly improved the quality of my research.

The work presented in this manuscript would not have been possible without the work and collaborations with PhD and master's students that I had the pleasure of supervising and co-supervising. Thank you for the work you have accomplished.

I am grateful to my colleagues and friends at the Ecole Nationale d'Ingénieurs de Carthage for creating a supportive and collaborative environment that has allowed me to thrive.

Finally, I would like to thank my family for their love and unwavering support. This accomplishment would not have been possible without their encouragement and belief in me.

Contents

Acknowledgements ii

List of Figures vi

List of Tables vii

1 General introduction 1

2 Texture classification 4
2.1 Introduction . . . 4
2.2 Binary classification based on automated fusion of baseline descriptors . . . 5
2.2.1 Motivation, Contributions and Overview . . . 5
2.2.2 Genetic Program Structure . . . 8
2.2.3 HOG and LBP Fusion . . . 9
2.2.4 Results and discussion . . . 11
2.3 Multi-class classification by learning texture descriptors . . . 12
2.3.1 Challenges and contributions . . . 12
2.3.2 Local Texture Description . . . 12
2.3.3 Global Texture Description and Classification . . . 17
2.3.4 Results . . . 20
2.3.5 Conclusion . . . 23

3 Facial expression recognition 25
3.1 Introduction . . . 25
3.2 2D facial expression recognition . . . 27
3.2.1 Motivation, contributions and overview . . . 27
3.2.2 Face detection and feature extraction . . . 28
3.2.3 Geometric Feature Extraction . . . 29
3.2.4 Texture feature extraction . . . 31
3.2.5 Learning binary programs by genetic programming . . . 32
3.2.6 Multi-class facial emotion recognition . . . 35
3.2.7 Results . . . 36
3.2.8 From 2D to 3D/4D FER . . . 38
3.3 3D/4D Facial Expression Recognition . . . 39
3.3.1 Motivation, contributions and overview . . . 39
3.3.2 Mesh-LBP calculation . . . 40
3.3.3 $Cov-3D-LBP$ matrices extraction . . . 41
3.3.3.1 LBP mean . . . 42

3.3.3.2 LBP difference . . . 42
3.3.4 Riemanian dictionary learning and sparse coding . . . 44
3.3.5 Classification methods used for 3D FER and 4D FER . . . 46
3.3.6 Results . . . 47
3.3.7 Discussion . . . 48
3.4 Conclusion . . . 49

4 Breast cancer diagnosis from mammographic images **51**
4.1 Introduction . . . 51
4.2 Literature review and proposed taxonomy . . . 53
4.3 Motivation and Contributions . . . 57
4.4 Local ROI texture representation . . . 58
4.4.1 Program representation . . . 60
4.4.2 Feature vector extraction . . . 61
4.4.3 Fitness function . . . 61
4.4.4 Genetic programming-based descriptor program generation . . . 63
4.5 Classification and ROI retrieval . . . 63
4.6 Results and discussion . . . 65
4.7 Discussion . . . 66
4.8 Conclusion . . . 69

5 Ongoing research and perspectives **70**

Bibliography **73**

List of Figures

2.1 A crystalline image from the DTD dataset: (a) Original image, (b) LBP transformation, (c) HOG transformation. 7
2.2 Flowchart of the proposed method for classifying texture images. 8
2.3 The program representation of an individual evolved by HL-GP. 9
2.4 HOG-LBP fusion process. 9
2.5 Edge detection and binarization process. 13
2.6 Distribution of 8×6 pixels around a central pixel P (8 orientations and 6 circles). . . 14
2.7 Vote weighting function for an edge pixel ($\sigma = 4$). 16
2.8 V_{EPO} and V_{EOH} for three texture instances, from two classes of the DTD dataset, using a distribution with $N = 4$ and $M = 8$ (4 orientations and 8 circles). 16
2.9 V_{EPO} and V_{EOH} of the same instance, from the DTD dataset, with two different scales using a distribution with $N = 8$ and $M = 8$ (8 orientations and 8 circles). 17
2.10 Overview of the proposed process for evolving a global descriptor and classifying texture. 18
2.11 Example of an Individual tree structure for a 3-bit code descriptor. 18
2.12 Feature vector generation: a 3-bit descriptor transforming the LES on a pixel to a vote in an 8-bit histogram feature. 19
2.13 Fitness measure as a function of the $^{DC}/_{HC}$ ratio. 20

3.1 Two faces displaying *disgust* emotional state that were miss-classified as *sadness* using geometric features according to a study in [134]. 26
3.2 Flowchart of the proposed method for 2D FER. 28
3.3 The set of 68 facial keypoints detected. 29
3.4 Two emotional states with approximately the same facial gestures: (a) anger, (b) fear. 29
3.5 An illustration of the lower lip elliptical shape during the display of (a) *happiness* (b) *neutral* and (c) *surprise*. 30
3.6 The representation of the eight facial ellipses. 31
3.7 The original face image and the the cropped image divided into 9 sub-images of size 40×40. 32
3.8 Overview of the proposed process for evolving a binary program. 33
3.9 Simplified tree representation of a binary program. 35
3.10 Filter tree representation. 36
3.11 Flowchart of the proposed method for 3D/4D facial expression recognition. 40
3.12 Construction of the face grid on the mesh: (a) the plane, formed by the tip of the nose and the two inner corners of the eyes, is defined, (b) an ordered and regularly spaced set of 35 points are calculated on the plane from the 3 landmarks, (c) the set of points is projected on the face surface, along the plane normal direction. 41
3.13 Illustration of the LBP mean. Five LBPs (a) and their mean $\hat{m}_I$ (b). 42

3.14 Euclidean distance *vs.* Hamming distance for comparing two LBPs (*e.g.* neighbors having a unit distance to the LBP '192'): (a) neighbors under Euclidean distance, (b) neighbors under Hamming distance. Numbers inside (*resp.* outside) parentheses denote the Euclidean (*resp.* Hamming) distances between the two connecting LBPs. 43
3.15 Structure of the HMM of an expressive sequence. 46
3.16 Frames extracted from a dynamic 3D video sequence illustrating the temporal dynamics of the happiness expression (the four states of the HMM are depicted in the sequence). 47
3.17 Comparison of the proposed method (PM) *vs.* state-of-the-art methods ([13], [46], [49], [60], [142] [83], [166], [7], [51], [161], [156] and [150]) for posed expressions in BU-3DFE dataset. 48

4.1 Benign, malignant and normal ROIs (of size 128×128) from the DDSM dataset and their corresponding LBP transforms and LBP histograms. 52
4.2 Flowchart of the proposed approach for breast cancer diagnosis from mammogram ROIs. 58
4.3 Overview of the offline phase for descriptor learning and knowledge base generation. . 58
4.4 Genetic process for generating discriminative features that enhance the classification accuracy. 59
4.5 Statistics extraction from a 3×3 window. 60
4.6 Example of a program tree structure for a 3-bit code descriptor. 61
4.7 Feature vector extraction: a 3-bit descriptor transforming the statistics on a local texture distribution to a vote in an 8-bit histogram feature. 62
4.8 Fitness measure as a function of the $^{D_c}/_{H_c}$ ratio. 63
4.9 The 10 first normal and abnormal ROIs retrieved (rank #1 marks the closest ROI and rank #10 marks the most distant ROI). 66
4.10 The 10 first malignant and benign ROIs retrieved (rank #1 marks the closest ROI and rank #10 marks the most distant ROI). 67

List of Tables

2.1 Terminal set of the GP-tree. 9
2.2 Average precision of existing methods compared to HL-GP (best results are in bold). . 11
2.3 A summary of the used benchmark image classification datasets. 21
2.4 Comparison of the proposed method (*GTS*), against relevant state-of-the-art methods, on five datasets while testing four different classifiers with 8×10 distribution using *Prot.I.* 22
2.5 Comparison with relevant state-of-the-art methods using *Prot.I* 23

3.1 The ellipses used to calculate the eccentricity features 31
3.2 Parameter setting of the genetic process. 35
3.3 Comparison with relevant FER methods using 10-fold cross validation on *DISFA+*, *CK+* and *MUG* datasets. 37
3.4 Comparison of the proposed method (PM) *vs.* state-of-the-art methods for posed expressions in BU-4DFE dataset. 49

4.1 Summary of the presented state-of-the-art methods. 56
4.2 Parameter setting of the genetic process. 64
4.3 A summary of the used benchmark mammographic ROI datasets. 65
4.4 Performance comparison against recent relevant state-of-the-art methods using the *2×5-fold* protocol on the DDSM dataset for abnormality detection. 68

Chapter 1

General introduction

As humans, we easily perceive the world that surrounds us, we swiftly detect the limit where texture changes. Visual perception is fascinating and at the same time intriguing. When looking at a flower garden through the window, we can infer the shape, the texture and the translucency of each petal through the subtle patterns of light and shadow that pass along its surface and effortlessly segment each flower from the background scene. Or, when looking at a framed group portrait, we can easily count and even name all the people in the photo, and even guess their emotions from their facial appearances. Perceptual psychologists have spent decades trying to figure out how the human visual system works. Even though they have been able to design optical illusions and unravel some of its principles, a complete solution to this conundrum remains elusive. Artificial intelligence and pattern recognition techniques have shown that it is not necessary to elucidate the mystery of natural vision to reproduce it on a machine. Image analysis is an active research areas that aims to extract meaningful information in some features and interpret them using different techniques. It can be as simple as reading a bar coded tags or as sophisticated as diagnosing cancer from medical images. Analysing images has attracted great attention from computer vision researchers due to its promising fields of application. This university habilitation dissertation falls within the field of research in image processing and computer vision. It presents a summary of my post-doctoral works and my contributions in this field, which has been carried out since 2014 within the research team "Systèmes Intelligents en Imagerie et Vision Artificielle" (SIIVA) of the LIMTIC laboratory. These works are the results of the supervision activities of young researchers and collaboration with senior members of the SIIVA team. Indeed, following my doctoral thesis in computer science, which was defended in December 2012, I became a member of LIMTIC laboratory and my research activities have expanded to various themes including texture classification, facial expression recognition and medical image analysis. Thus, the research work that I conduct concerns mainly three axes of research.

The first axis revolves around texture image classification, which is an important topic of computer vision that has been applied to a wide variety of applications such as facial emotion recognition, pedestrian detection and medical image processing. This topic continues to attract many researchers trying to solve multiple problems. Indeed, analyzing image content in order to define representative information remains very challenging and can greatly improve classification results. One of the most important steps in image classification is the feature extraction process. It refers to the extraction of representative information from an image in order to describe data while reducing dimensionality. Three types of descriptions are commonly used: color, shape and texture. The latter being an important visual pattern composed of entities with a homogeneous spatial organization. Since texture is considered as a highly informative pattern in various applications, many descriptors have been

developed in order to analyze textured images efficiently. However, existing methods faced multiple challenges when learning a texture classifier. On the one hand, it can be difficult to provide a reliable dataset in order to detect and extract high level features. The training process requires a big number of instances that cannot be always available in real-world scenarios. Besides, even when labeled data are provided, the detection and extraction processes may grow in complexity and the trained model can easily fall into overfitting. On the other hand, it is very difficult and time consuming to design a feature extraction method without the need of human intervention. To tackle these problems, some methods have focused on automating the feature detection and extraction process. In this perspective, we proposed to automate the process of texture classification while describing texture locally and globally to guarantee robustness when facing illumination and geometric changes. To achieve this, first, we proposed a Genetic Programming (GP)-based method that combines the two well-known features of histograms of oriented gradients and local binary patterns. Indeed, a three-layer tree-based binary program is learned using genetic programming for each pair of classes. The three layers incorporate patch detection, feature fusion and classification in the GP optimization process. The feature fusion function is designed to handle different variations, notably illumination and rotation, while reducing dimensionality. Second, we proposed a new operator, which we named Local Edge Signature (LES) descriptor, to locally represent texture. The proposed texture descriptor is based on statistical information on edge pixels' arrangement and orientation in a specific local region, and it is insensitive to rotation and scale changes. A genetic programming-based approach is then fitted to automatically learn a global texture descriptor that we called Genetic Texture Signature (GTS). In fact, a tree representation of individuals is used to generate global texture features by applying elementary operations on LES elements at a set of keypoints, and a fitness function evaluates the descriptors considering intra-class homogeneity and inter-class discrimination properties of their generated features.

The second axis is about facial expression recognition which is also a major field of research in computer vision and pattern recognition. The growing interest in the analysis of human faces comes not only from its ability to reveal demographic information (gender, age, ethnicity, etc.) or the person's identity, but also because it is considered as an important emotional and awareness communication channel, which reflects some of our cognitive activities and well-being. In fact, people from different cultures show the same facial expressions for the same feelings. The strong acceptance of this affirmation in psychology motivated researchers in computer vision and affective computing to develop automated systems for emotional states and human affects detection and understanding from facial expressions. Facial features analysis started with 2D still images. Many applications were realized, such as facial expression recognition (FER) under rigorously constrained conditions. In this context, the first work within this axis concerns 2D FER. Indeed, a multitude of features have been proposed in the literature to describe facial expression. None of these features is universal for accurately capturing all the emotions since facial expressions vary according to the person, gender and type of emotion (posed or spontaneous). Therefore, some research works have considered combining several features to enhance the recognition rate. But they faced significant problems because of information redundancy and high dimensionality of the resulting features. Therefore, we proposed a genetic programming framework for feature selection and fusion for 2D facial expression recognition, which we called $GP-FER$. The main component of the proposed framework is a tree-based genetic program with a three functional layers (feature selection, feature fusion and classification). The proposed genetic program is a binary classifier that performs discriminative feature selection and fusion differently for each pair of expression classes. The final emotion is captured by performing a unique tournament elimination between all the classes using the binary programs. Three different geometric and texture features were fused using the proposed $GP-FER$. Nonetheless, the poor performance presented by 2D still images in FER leads to a deficiency in the temporal information. Besides, problems;

like illumination variation, head pose variation, occlusions and scale variation; reduce dramatically the performance of 2D FER in real-world scenarios. Thus, the second work presented in the second axis focused on 3D/4D FER. Indeed, an effective method for automated 3D/4D facial expression recognition based on Mesh-Local Binary Pattern Difference (mesh-LBPD) is presented. In contrast to most of existing methods, the proposed mesh-LBPD is based on a unified set of geometric and appearance features of different facial regions. Multiple features are combined into a compact form using covariance matrices, namely $Cov-3D-LBP$. Then, the $Cov-3D-LBP$ atoms are represented sparse data combinations. To that end, a Riemannian optimization objective for dictionary learning and sparse coding is used, in order to reduce the complexity of the problem, and the representation loss is characterized via an affine invariant Riemannian metric.

The third axis focuses on medical imaging and more specifically in the diagnosis of breast cancer from mammographic images. In fact, according to the World Health Organisation, breast cancer causes about 15% of cancer deaths. Mammography is the first common standard for routine screenings for breast cancer since it is a fast and affordable technique. It aims to reduce the mortality by detecting breast cancer at an early stage even before woman feel the symptoms. In this stage the breast cancer is easily treatable and the risk of fatality is low. However, according to radiologists, there are some shortcomings faced when mammography is used as the only radiological tool in order to assess a patient's risk for breast cancer. Indeed, cancer cell tissue at an early stage is difficult to differentiate from breast tissue. The presence of dense breast tissues (parenchymal tissue) in the breasts of some patients complicates the expert diagnosis, which results in false negative diagnoses of mammograms for those patients having dense breasts. Our work within this axis aims to understand the challenges facing breast cancer diagnosis systems and to propose an accurate and automatic framework for breast cancer detection. Thus, the first work aims to present a taxonomy of the most relevant works within the framework of breast cancer diagnosis from mammographic. A brief and exhaustive literature review is presented and works are classified in categories and subcategories. Pros and cons of each sub-categories is then summarized in a tabular way based on the results and discussion presented in the original papers. The second contribution is the suggestion of a fully automated method for local feature extraction and global feature generation for mammogram images. To achieve this end, a local breast tissue representation is proposed, and a genetic programming-based descriptor is designed to transform local features into a global one. The evolutionary process is based upon a fitness function that guarantees the discriminative power of the descriptor while using small training instances. Indeed, analysing local texture and generating features are two key issues for automatic cancer detection in mammographic images. Recent researches have shown that deep neural networks provide a promising alternative to hand-driven features which suffer from curse of dimensionality and low accuracy rates. However, large and balanced training data are foremost requirements for deep learning-based models and these data are not always available publicly. Thus, we propose a fully-automated method for breast cancer diagnosis that performs training using small sets of data. Feature extraction from mammographic images is performed using a genetic-programming-based descriptor that exploits statistics on a local binary pattern-like local distribution defined in each pixel.

The remainder of this manuscript is organized as follows. Chapter 2 reports contributions in the topic of texture classification. Chapter 3 addresses the topic of facial expression recognition and presents contributions in 2D and 3D/4D FER. Chapter 4 presents the contributions within the framework of breast cancer diagnosis from mammographic images. Chapter 5 synthesizes my current research and presents some perspectives for future directions.

Chapter 2

Texture classification

The main results presented in this chapter have been published in the following international journals: Expert Systems With Applications (ESWA 2020) [39], The Visual Computer (TVC 2021) [54] and EVOlving Systems (EVOS 2021) [53] in addition to the main conference: Engineering Applications of Neural Networks (EANN 2020) [42].

2.1 Introduction

Texture description and classification play a fundamental role in many computer vision applications, such as surface inspection of materials, medical imaging, image recovery, object and scene recognition. Due to the importance and the abundance of application fields, several texture classification approaches have been proposed during the last years. However, extracting highly representative and robust texture characteristics to describe textured images is a difficult problem that needs further investigation. In fact, many classical descriptors have proven to be efficient when describing texture. For instance, edge-based descriptors have focused on calculating first-order or second-order statistical measures of edge distributions. Indeed, gradient magnitudes and directions are calculated in a local neighboring in order to extract the edge distribution. Other descriptors focused on spatial frequencies of texture primitives. For example, the Local Binary Pattern (LBP) descriptor and its variants tried to model local texture by calculating the difference between a central pixel intensity and its neighbors' intensities within a local support region. All these descriptors tried to describe local texture by means of features based on the occurrence of corners, the direction of edges, the shape of textures, the difference of pixel intensities and/or the spatial frequencies of texture primitives. Although some of these descriptors have managed to achieve good results, they still suffer from weakness against textures with scale, rotation or/and deformation variations. The most likely causes of this weakness are that these descriptors lacked to model the texture as seen by human visual sense and also focused very locally while forgetting that texture is defined in a global way. Thus, many textural features were defined to mimic the way human perceive texture, such as coarseness, contrast, complexity, busyness, shape, directionality and texture strength. Indeed, some works [115, 12] compared the computational with human ranking for the texture classification, and they stated that there was a good correspondence between the two. A major disadvantage of almost human-like approaches is that they do not have general applicability. The human perception mechanism, in comparison, seems to work well for almost all types of textures. There is some agreement between most researchers about the main categories of texture classification, but they also note that humans tend to combine rather than use one single method. In [115], authors have carried out sets of human studies, where volunteers were asked to select which images, in order, from a set of 111 images, they considered to be most like a given target image. Figure 3.4.a shows a target

image and Figure 3.4.b refers to the most like image according to human subjects. Most proposed descriptors in literature would not be able to classify these two images as being of the same class. Indeed, these descriptors are totally decorrelated from the way humans perceived the resemblance between the two textures. In fact, the correlation between the two textures in Figure 3.4 lies in the curvatures of the contours and the frequency of the discontinuities along the different directions. Texture is a complex visual pattern composed of entities with homogeneous spatial organization [92]. For a human being, it is easy and sometimes even natural to discriminate textures present in an image. Nevertheless, for a machine, there are several factors to take into account in order to make a precise classification. Indeed, texture images are often taken in uncontrolled environments and their treatment requires expert's intervention in order to select and extract useful and robust texture features. Texture classification relies heavily on two main steps, namely primitive detection and texture feature extraction, which have a direct impact on the conception of the classifier. Even if the two aforementioned tasks have been used interchangeably, their goals remain different. In fact, primitive detection aims to identify patterns (*e.g.* regions, contours, patches, points...) on which the texture will be defined [67], whereas texture feature extraction aims at transforming raw pixel values of a detected primitive, or its surrounding region, into a reduced domain [143, 80]. Overall, the success in performing a good classification is highly dependent on the quality of the extracted texture features. However this process depends on domain knowledge of the task and can be highly costly. These challenges have attracted increasing attention and many image descriptors have been introduced in order to analyze textured images efficiently while extracting high level texture characteristics. Motivated by the success of several evolutionary techniques in several image classification tasks [PUT SOME REFERENCES], we propose to use Genetic Programming to fully automate texture classification. A binary classification based on the fusion of HOG and LBP descriptors is, first, proposed in section . An approach to learn texture descriptor for multi-class classification is then explored in section .

2.2 Binary classification based on automated fusion of baseline descriptors

2.2.1 Motivation, Contributions and Overview

Texture classification is a challenging problem for many computer vision applications such as surface inspection of materials, medical imaging, image recovery, remote sensing imaging and object recognition [59, 64, 29, 163]. Two of the most used descriptors for texture classification are the Local Binary Patterns (LBP) [111] and the Histogram of Oriented Gradients (HOG) [24]. On the one hand, LBP is a local descriptor that considers the eight neighbors of a central pixel and computes the difference between their values. A feature vector is then constructed with the patterns being the feature index, and the feature value being the number of pixels having the pattern. On the other hand, HOG calculates the gradient vector for each pixel in order to define a histogram of eight bins. In fact, the gradient magnitude is defined for each pixel towards the total value of the corresponding bins. Figure 2.1 shows the HOG and LBP transformations of a crystalline image taken from the widely used Describable Texture Dataset (DTD) [20]. We can notice the high discriminative power of LBP, which reveals very effective to capture local patterns exploiting the eight directions of each pixel (Figure 2.1(b)). However, its binning coarseness makes it lose information compared to HOG. In fact, HOG performs better in capturing edges and corners considering the direction for which the gradient magnitude is the greatest (Figure 2.1(c)). Many studies [107, 152] have used LBP as a descriptor for texture image classification while achieving good results, specially with images presenting a scale variation.

However, they show poor results when it comes to rotation and/or photometric deformations such as illumination and noise. The most likely cause of this weakness resides in the fact that these LBP-based methods lacked to model the texture as perceived by human visual sense. Besides, they are focusing very locally while forgetting that a texture is also defined in a global way. Furthermore, far less studies have focused on HOG as a texture descriptor due to its inability to distinguish between local minimums and maximums, even if it shows more robustness when it comes to rotation. Thus, few studies have tried to combine both descriptors to exploit their complementarity [113]. Most of these methods suffer from further problems such as large feature vectors, long computation time, redundancy and the need for human intervention.

Therefore, two questions arise. First, is it possible to merge the HOG and LBP descriptors without redundancy and without exploding the computing time? Secondly, is there a way to detect representative texture characteristics without human expert intervention? To deal with these issues, this paper attempts to benefit from the prior designed image-related operators and descriptors in order to develop a Genetic Programming (GP)-based method for effective texture image classification. Indeed, we propose a cost-efficient method, named HL-GP (H and L stand for HOG and LBP, respectively), that combines a set of image-related functions and terminals, inspired by HOG and LBP features, while using genetic programming. These functions, which operate directly on raw pixels, are designed to detect more informative and advanced features than the existing GP-based methods. In fact, our GP-tree structure is composed of multiple layers while including a feature construction process that is missing in most GP-based methods. This allows the proposed method to simultaneously perform patch detection, feature extraction and image classification. The main contributions of this work can be summarized as follows. A function set that combines HOG and LBP descriptors is proposed in order to automatically produce high-level features for classifying texture images. The final output is a low dimension feature vector, unlike most outputs generated by common fusion techniques [61, 62]. It gives the proposed method the potential of achieving accurate classification on challenging datasets of texture images involving both photometric and geometric changes. Moreover, we introduce a program representation that can integrate these functions in order to perform patch detection, feature extraction and classification in a single GP tree. The performance of the proposed fully automated classification method is examined and compared with several relevant state-of-the-art GP and non GP-based methods, on six challenging texture datasets, for both binary classification.

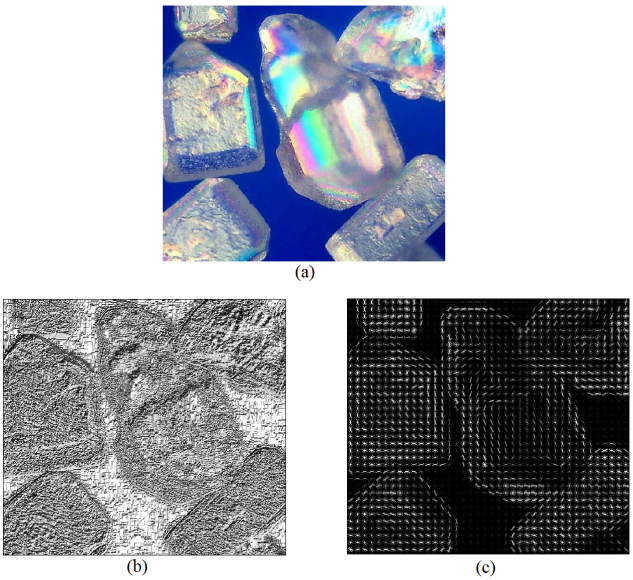


Figure 2.1: A crystalline image from the DTD dataset: (a) Original image, (b) LBP transformation, (c) HOG transformation.

Most of the existing methods that have been designed to combine HOG and LBP descriptors involve human intervention, *i.e.* they need human experts to select discriminatory keypoints and/or manually setup descriptors to extract features. Furthermore, existing methods use complex operations, high number of training instances and greedy computing processes. It is worth highlighting that, even if GP-based methods allow the automating of the general process, no study has proposed a solution for combining HOG and LBP for texture classification, as well as we know. This is probably due to the high dimension of feature vectors generated by common concatenation and fusion techniques, what requires a feature selection step in order to reduce the dimensionality. Therefore, all mentioned GP-based methods have focused only on using each descriptor separately. Differently, this work proposes a method that uses GP to tackle all these issues while performing patch detection, feature extraction as well as image classification, using a fusion function that combines HOG and LBP features. Figure 2.2 shows the different steps of the overall algorithm. First, during the training phase, data is pre-processed and unfit images are cropped to match dimensions set on the program parameters. Then, instances are fed to the GP process to evolve classifiers which are evaluated using the fitness function. If the stop criterion is met, the algorithm returns the best evolved program representing an automatically evolved image classifier. Otherwise, the GP process is run again. The evolutionary process measures the quality of a program using the fitness function. Indeed, a program's performance is correlated with its ability to correctly classify training instances. Therefore, an ideal program will have an ideal fitness value of n corresponding to the total number of training instances. The minimum value is 0 representing the worst possible outcome (if the program does not classify any image correctly). The GP classifier, which is based on a tree structure, is evaluated in a bottom-up manner producing a single value as output. A value less than zero associates an image to the negative class (0), whereas a value greater than zero associates the image to the positive class (1). The program with the highest fitness is retained as the best classifier for the two classes in question and will be tested thereafter. In fact, during the test phase, the remaining of instances are used in order to predict a class label and assess the performance of the evolved classifier on unseen data.

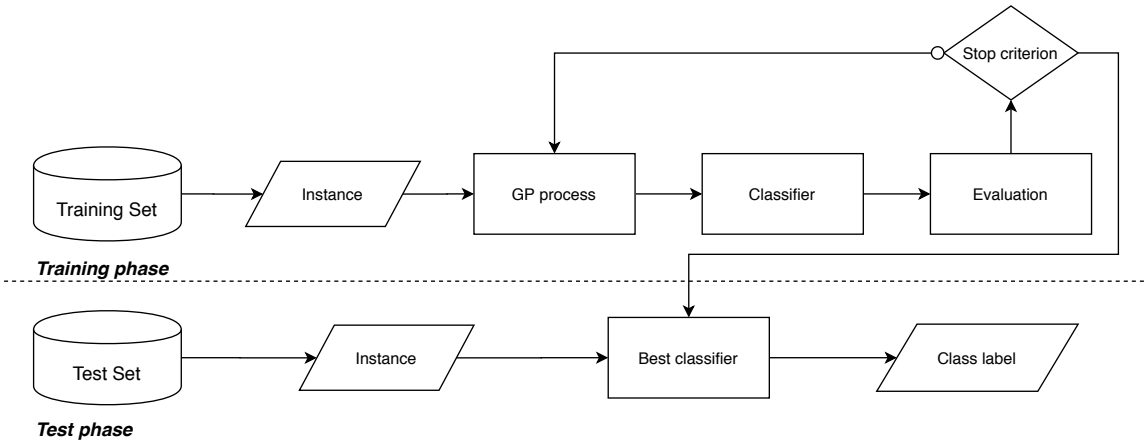


Figure 2.2: Flowchart of the proposed method for classifying texture images.

2.2.2 Genetic Program Structure

The main objective of the algorithm is to achieve patch detection, feature extraction and classification simultaneously. For this purpose, a multi-layer GP structure is designed. Figure 2.3 depicts an example to show how the layers of an evolved individual are constructed. It is a tree-based GP representation where the operators are the internal nodes and the terminals are the leaf nodes. To introduce restrictions on inputs and outputs of the different nodes, strongly-typed GP [105] is used. The program is composed of three layers: Patch Detection (PD), Texture Feature Extraction (TFE) and Classification. The proposed program structure includes also a construction process, which is missing in existing GP-based classification methods, in order to provide high level features from the low level ones. The full terminal set and its details are listed in Table 2.1. The bottom layer (PD) includes three terminal functions which are the image to be evaluated, the size and the position of a rectangular patch. The "image" node is a 2D array composed of the intensity values of input pixels. The "position" node is a pair of values $(x,\ y)$ that define the coordinates of the upper left corner pixel of a random generated rectangular patch. The "size" node defines the size of the patch and is composed of a pair of values (width, height). The position and the size of patches are generated randomly within the input image boundaries for each random program throughout the evolutionary process. In fact, working on patches of the image decreases the computing time and allows the program to detect regions containing important keypoints. The mid-layer (TFE) performs feature extraction from the selected patches. This layer is composed of a set of functions, including the node "index" which returns a value between 0 and 10 since the generated histograms are composed of 11 bins. It also includes the "bin", "HOG-LBP" and "distance" nodes which are discussed in the next subsection. A construction process is then performed on the previously obtained features in order to obtain high level features using several functions (bin, distance, index, add...). Finally, the upper layer (Classification) assigns a class label to the input image after comparing the value obtained with a predefined threshold. This layer includes the four arithmetic operators "sub" (−), "add" (+), "mul" (×) and "div" (/). They take two arguments as input and return a single output which can be used as input for the parent node. The "div" operator is protected to avoid the "division by zero" problem, by returning zero whenever the denominator is equal to zero. The arithmetic operators in the function set have their corresponding regular meanings and allow GP to use multiple extracted features for classification. Generally, the program tree is built in a bottom-up manner and necessarily contains the three layers. Leaf nodes (image, position, size) are defined to perform patch detection. However, HOG-LBP, index, bin, distance, add, sub, mul and div functions are defined to perform

feature extraction and classification. An in-depth analysis of the proposed HOG and LBP fusion is provided in the next subsection.

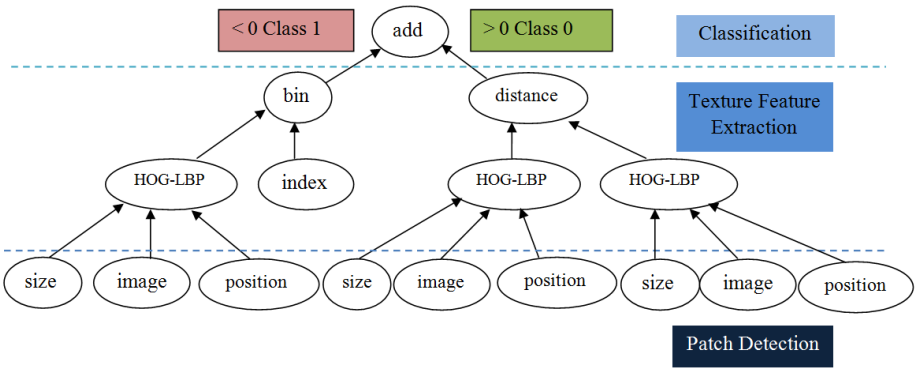


Figure 2.3: The program representation of an individual evolved by HL-GP.

	Details
image	2D array representing the image to be classified
position	Position $(x,\ y)$ of a patch of the image, where x is the horizontal location randomly generated in $[1,\ MinWidth]$ and y is the vertical location randomly generated in $[1,\ MinHeight]$
size	A random vector generated in $[3 \ldots MaxWidth,\ 3 \ldots MaxHeight]$
index	A random integer generated in $[0,\ 10]$ representing a bin index of a histogram

Table 2.1: Terminal set of the GP-tree.

2.2.3 HOG and LBP Fusion

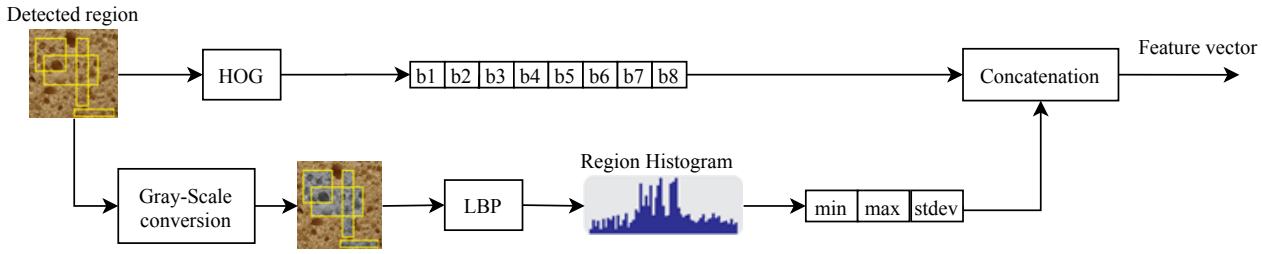


Figure 2.4: HOG-LBP fusion process.

The proposed HOG-LBP fusion mechanism exploits advantages from HOG and LBP descriptors by trying to combine both while taking into account the different deformations the image may face. Indeed, for a descriptor to be robust, it must take into account different variations such as rotation, scale and illumination. Another problem yet to be addressed is how to combine the two descriptors knowing that they capture information differently. Even if both descriptors are based on the gradient information, each one processes in a different manner and produces a different output. HOG proves very effective when it comes to capturing the outlines and angles, and it generates, in our case, a histogram of 8 bins. LBP on the other hand uses 8 directions for each pixel and generates a 256 bin

histogram representing the pixel distribution of the input image. However, as mentioned by Ojala [110], for a histogram composed of 256 bins, only a small set is relevant and using prominent bins like minimum and maximum can enhance the discrimination power since it can capture differences between one texture and another. The concatenation of both vectors is a solution to form a single feature pool, but in order to limit the number of features while avoiding the curse of dimensionality, we designed a GP-based fusion technique (Figure 2.4). The length reduction does not affect a classifier's performance. In fact, each region is selected according to the input arguments of the "HOG-LBP" function which are the position and the size. Afterwards, a rectangle patch is defined and two algorithms are executed simultaneously. The first part is inspired from HOG but differs slightly from the standard version in order to allow it to be compatible with GP. Indeed, the designed algorithm is not applied on the whole image. Hence, it does not use multiple overlapping blocks to obtain a feature vector as in the standard version. Instead, the GP tree automatically analyzes the image and generates several patches. A feature vector is then computed for each patch. The second part is inspired by LBP and also differs from the standard version since it is not applied on the whole image. It works on a single block of variable size. Thus, it generates a histogram, of 256 values, which is then normalized and transformed into a three dimensional feature vector composed of its minimum, maximum and standard deviation. The intuition behind choosing these functions is their order-independent property when extracting features. In other words, shuffling the values of the vector will not affect the results returned by those functions. This is very important to handle the rotation variants of the pixels. Another difference from standard approaches that is common to both parts of the algorithm is the normalization of the histograms, which is now applied separately for each selected patch. The normalization makes it possible to fuse HOG, which is a global feature, and LBP, which operates in a local manner, easily without the need for a feature selection method since the two histograms have the same weight. At the end of the process, the two histograms are concatenated to form the final vector. Unlike other HOG and LBP combination methods, the HL-GP fusion process does not take into account redundant information. Indeed, the proposed fusion method captures the gradient orientations given by the HOG descriptor in addition to three discriminative statistics (*min*, *max* and *stdev*) computed from the LBP descriptor. The goal behind reducing the LBP feature vector dimension is to minimize the risk of selecting unnecessary or redundant features, which can alter the classifier's performance. It is also worth mentioning that the feature selection is incorporated in the genetic learning process, which guarantees the extraction of the most prominent and non-redundant set of features from both HOG and LBP descriptors thanks to the GP optimization. The output is composed of 11 bins which cannot be used directly by GP for classification purpose. Therefore, two other functions are added to overcome this problem. First, the "bin" function that returns the value of a given bin selected by the "index" node. It allows the GP tree to discriminate histogram values (max, min, stdev or one of the 8 orientation bins). Second, the "distance" function makes it possible to assess the distance between two histograms. Indeed, distance can be an important feature for classification as generated patches can give very different histograms depending on pixel values. Through the learning process, three distance metrics have been investigated, and the one with the best learning rates is then adopted for our classification problems. The first one is the Euclidean distance, which is the most obvious way of representing distance [25]. The second distance is the Cosine distance, which is a measure of similarity between two non-zero vectors of an inner product space [108], and it defines the cosine of the angle between them. The third distance is inspired from the Chi-squared test, which is commonly used for testing relationships between categorical variables [133].

	Method	Texture 1	Texture 2
Non-GP methods ($mean \pm \sigma$)	SVM [139]	59.8 ± 22.6	48.9
	NB [127]	67.1 ± 13.5	-
	NB Tree [130]	68.5 ± 10.6	-
GP-based methods ($mean \pm \sigma$)	Conv-GP [35]	55.4 ± 6.6	-
	3TGP [6]	-	82.7 ± 4.2
	uLBP+GP [110]	-	92.4 ± 2.8
	2TGP [132]	51.8 ± 2.1	75.6 ± 3.9
	GP-HOG [84]	75.5 ± 6.2	76.4 ± 3.2
	GP-GLF+1-NN [15]	-	**97.8 ± 0.7**
	HL-GP	**82.1 ± 3.7**	87.2 ± 2.5

Table 2.2: Average precision of existing methods compared to HL-GP (best results are in bold).

2.2.4 Results and discussion

For our experiments, visually similar classes are selected. Indeed, two classes from the KTH-TIPS dataset are used to form the "Texture 1" collection, and two other classes are drawn from the KTH-TIP2a dataset to form the "Texture 2" collection.

Table 2.2 shows the results in terms of mean and standard deviation of the classification accuracies obtained by HL-GP and other standard methods on "Texture 1" and "Texture 2". On the "Texture 1" collection, HL-GP largely outperforms the non-GP methods (*i.e.* SVM classifier, NB and NBTree) with an improvement exceeding 20% compared to SVM and 14% compared to NB and NBTree. For methods using genetic programming, Conventional-GP and 2TGP show poor results and achieve 55.4% and 51.8% as accuracy, respectively, while GP-HOG reaches an accuracy of 75.5% but remains significantly lower than the proposed method. For the "Texture 2" collection, the performance of HL-GP remains superior to the SVM classifier, which only achieves 48.9% as accuracy rate, as well as to the other GP methods: 3TGP with 82.7%, 2TGP with 75.6% and GP-HOG with 76.4%. However, the uLBP + GP method records better results with an average accuracy of 92.4%. Indeed, the images in this collection contain a large amount of texture information that can be well captured by the uniform LBP histogram features of this method. Unlike HL-GP that extracts features from the detected patches, the uLBP + GP works over the entire image, what could further improve its performance. The best results are achieved by GP-GLF with almost 98% accuracy rate and a low standard deviation (σ). However the GP architecture of this method does not perform image classification. It is combined with a 1-NN and its performance may change with the use of other classifiers. GP-GLF also uses a huge number of instances for training as it needs 50% of the dataset to train the evolved descriptor and another 25% for 1-NN training leaving only 25% of unseen data, what makes it subject to overfitting. It is worth noting that this comparison does not take into account the performance of other methods when using a limited number of instances. Indeed, HL-GP evolves a classifier with only 15 training samples, what allows to use the rest as a test set while the other methods use half of the dataset for training and the other half for the test.

2.3 Multi-class classification by learning texture descriptors

2.3.1 Challenges and contributions

Texture classification approaches still need to deal with the following two main challenges: how to describe locally a complex texture with relatively low dimensional measures while remaining insensitive to changes that may occur? and, how to aggregate these local texture measures to obtain a global texture description? The contributions of this work come to deal with these two issues. Firstly, a new local texture measure, named Local Edge Signature (LES), is proposed to describe local texture with a 6-dimensional vector. This local texture representation uses statistics on structural information in a specific local distribution of pixels around a central pixel. The local distribution is designed to have rotation and scale insensitive statistical information describing the local texture. Thus, the proposed LES descriptor tries to combine different aspects of the human way to perceive texture in a low-computational feature that has general applicability. Secondly, a genetic programming approach, which we called GTS for Genetic Texture Classification, automatically evolves robust texture descriptor from a small number of training instances. In fact, to obtain a global feature, tree representation using arithmetic and comparison operators is proposed in order to aggregate local edge signatures on a set of keypoints. Furthermore, a fitness function considering the intra-class homogeneity and inter-class discrimination properties of the features is designed. The significance of the proposed method lies in the generated descriptor that extracts discriminative and geometric-insensitive texture features from a small training set, without the need for an expert intervention. This makes it particularly appropriate for expert and intelligent systems dealing with numerous real-life problems of society and industry that include unconstrained content-based image classification [141] and retrieval [97], biomedical image analysis [66] and multi-spectral remotely sensed imaging [149]. In what follows a detailed description of the proposed method for texture description and classification is given. Indeed, after presenting the process of extracting the local texture signature descriptor, we give an overview of the proposed genetic texture classification algorithm including coding of the genetic individuals, feature vectors' extraction and fitness calculation.

2.3.2 Local Texture Description

The goal herein is to define local texture features for a pixel as independently as possible from the rotation and the scale. In this work, the texture is described locally, by the dispersion of edge pixels around a central pixel in a specific neighborhood (defined by a distribution matrix). For this purpose, a step of edge detection is needed in order to binarize the input image while obtaining an edge image, in which the edge pixels are labelled by 1 and non-edge pixels by 0. This edge image is used to binarize the local distribution matrix in order to obtain a matrix describing the dispersion of the edge pixels through a number of orientations crossing the central pixel. Statistics on edges' orientations and their spatial frequencies around the central pixel will be then extracted in a vector called Local Edge Signature (LES). More details about the extraction of LES will be given after describing the process of edge detection and image binarization.

In order to extract texture information, a step of edge detection and image binarization is needed (Figure 2.5). In fact, since texture can be accurately described by edges, edge detection is one of the most commonly used operations in image analysis within the framework of texture classification [128, 3, 114, 140, 74]. An edge is defined by a discontinuity in intensity values. Many filters are used in the process of identifying the image edges by locating intensity discontinuities. For instance, gradient

filters are first derivative filters used for edge detection, and Laplacian filters are second derivative filters used to find areas of rapid changes (edges) in images. However, since derivative filters are very sensitive to noise, it is common to smooth the image using a Gaussian filter before applying the Laplacian. This two-step process is called the Laplacian of Gaussian (LoG) operation. Indeed, fine edges are detected while combining Laplacian of Gaussian and gradient filters. A 5×5 convolution filter (2.1) is used as LoG filter and $M_y = [-1, 0, 1]$ and $M_y = [-1, 0, 1]^T$ filter kernels are used to compute the gradient magnitudes and orientations. In fact, three standard convolution filter sizes (*i.e.* 3×3, 5×5 and 7×7) have been investigated, for a sample of texture images, while evaluating the quality of the resulting edge images. Obtained results have proven that the 5×5 size ensures the trade-off between the accuracy of edge localization and the computational time cost. Then, the input image, already treated by the LoG filter, is firstly fed to a zero-crossing detector in order to generate the LoG-based binary edge image (edge pixels *vs.* non-edge ones). A second gradient-based binary edge image, with same labelling as the first one, is thereafter generated using a thresholding process on pixel magnitudes (magnitude peaks detection). The final binary edge image E is then obtained by applying the binary AND operator on the two binary edge images. Pixels labelled by 1 in the binary edge image E correspond to pixels with zero crossings in the LoG image and peaks in gradient magnitudes. The image refined by gradient filters is used to generate a magnitude matrix M and an orientation matrix O. Indeed, two partial derivative images I_x and I_y are defined by convolving the input image I with the two gradient filters M_x and M_y, respectively (*i.e.* $I_x = M_x * I$ and $I_y = M_y * I$). The matrix M is calculated by estimating the magnitude in each pixel, which is equal to $\sqrt{I_x^2 + I_y^2}$. However, the orientation matrix O consists of the orientation in each pixel, which is estimated as $\arctan(I_y/I_x)$. Let $edge()$ be the function that corresponds to the matrix E. Given a pixel p, $edge(p)$ returns 1 if p is an edge pixel and 0 otherwise. In the next section, the process of extracting the proposed Local Edge Signature descriptor (LES) will be detailed.

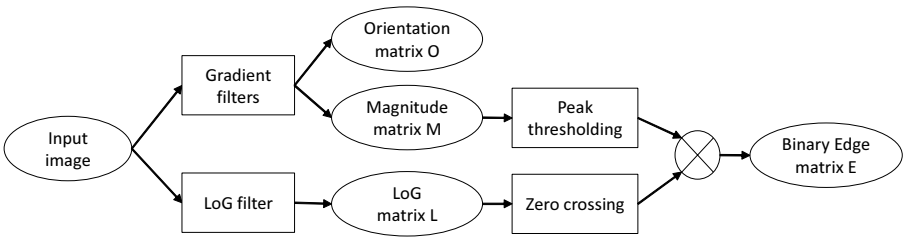


Figure 2.5: Edge detection and binarization process.

$$\nabla^2 g(x, y) \approx \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}. \tag{2.1}$$

In order to locally describe texture in a pixel of the image, a support region has to be specified. In this work, a specific neighborhood, denoted by $D_{N,M}(P)$, is defined around a pixel P. It refers to a distribution of $N \times M$ pixels scattered around the pixel P. $D_{N,M}(P)$ is defined by N orientations $\theta_i = \frac{2(i-1)\pi}{N}$ ($i \in \{1, \ldots, N\}$) and M circles C_j ($j \in \{1, \ldots, M\}$). Each circle C_j is centered at the pixel P and has a radius of j pixels. The distribution $D_{N,M}(P)$ is characterized by its number of orientations N (step angle of $\frac{2\pi}{N}$) and its number of circles M. Figure 2.6 shows a distribution with $N = 8$ (8 orientations) and $M = 6$ (6 circles). The size of a local distribution is referred to with

$N \times M$. Let P_{ij} be the pixel at the intersection of the orientation θ_i and the circle C_j, the distribution $D_{N,M}(P)$ is defined by the $N \times M$ matrix given by (2.2):

$$D_{N,M}(P) = \begin{matrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_N \end{matrix} \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1M} \\ P_{21} & P_{22} & \cdots & P_{2M} \\ \vdots & \vdots & \vdots & \vdots \\ P_{N1} & P_{N2} & \cdots & P_{NM} \end{bmatrix}. \tag{2.2}$$

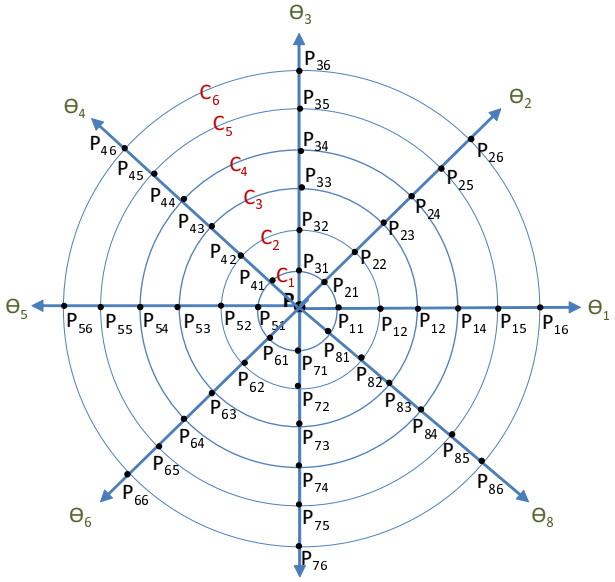


Figure 2.6: Distribution of 8×6 pixels around a central pixel P (8 orientations and 6 circles).

Let ψ be a scalar function defined on the image pixels, incarnating either a primitive (*e.g.* magnitude, orientation) or photometric (*e.g.* color or gray level) information, the matrix $D_{N,M}(P,\psi)$ obtained by applying ψ to every element of $D_{N,M}(P)$ is defined as follows (2.3):

$$D_{N,M}(P,\psi)(i,j) = \psi(P_{ij}) \ . \tag{2.3}$$

We define the Edge Signature matrix $ES(P) = D_{N,M}(P, edge)$ at the pixel P, as the distribution of edge pixels around P. In other words, $ES(P)$ is the matrix obtained after binarizing the $D_{N,M}(P)$ matrix by replacing each pixel element by 1 if it is an edge pixel and 0 otherwise. Thus, ES is used to generate two local feature vectors, which are EPO (Edginess Per Orientation) and EOH (Edge Orientation Histogram). EPO is an N-dimensional vector (N is the number of orientation in $D_{N,M}(P)$), such that each EPO element is the quotient of the number of edge pixels in the corresponding orientation by the total number of edge pixels (2.4). Hence, the vector EPO, which describes the frequency of texture edges through the different orientations, significantly informs about the nature of the texture (line-like texture, regular texture, random texture, rough texture...).

$$EPO(P)(i) = \frac{1}{\sum_{m=1}^{M}\sum_{n=1}^{N} ES(m,n)} \sum_{j=1}^{M} ES(i,j). \tag{2.4}$$

EOH is a histogram feature defined in each pixel P and is built using the orientations of edge pixels within the local distribution $D_{N,M}(P)$. Indeed, each edge pixel within the $D_{N,M}(P)$ region casts a weighted vote with his gradient value (given by the matrix M) for an orientation histogram channel

based on its orientation (given by the matrix O). Unsigned orientations are used so the histogram channels are spread over 0° to 180° and discretized into nine histogram channels with an angular step of 20°. For the vote weight, pixel distance to the center is taken into consideration for the pixel contribution. Moreover, the Gaussian weighting function (2.5) is used for the pixel vote. To account for changes in illumination and contrast, the gradient strengths are normalized by dividing each bin by the sum of all histogram bins.

$$w(P_{ij}) = edge(P_{ij}) \times e^{\frac{-(j-1)^2}{2\sigma^2}}, \tag{2.5}$$

where, σ is a scaling factor that was set using an intuitive heuristic. This heuristic divides the pixels within the local distribution into two sets of equal numbers based on their distances to the center pixel: the closet pixels and the most distant ones. The weights are assigned so that pixels within the set closest to the center have weights greater than 0.5, and vice versa for pixels in the other set. This heuristic has allowed local texture discrimination power when examining several EOH features for same and different class instances, even with scale changes. For a local distribution with 10 circles ($M = 10$) for example, the value of σ was set to 4 . As shown in Figure 2.7, the fifth closet pixel to the center has a weight of 0.6 however the sixth closet one has a weight of 0.42. The reason behind the use of such a function is to give more consideration to the vote of edge pixels near the central pixel in order to minimize the effect of the scale on the final histogram. Once EPO and EOH features have been calculated for a pixel P, we extract statistical information, in terms of two 3-dimensional vectors V_{EPO} (2.6) and V_{EOH} (2.7), as follows:

$$V_{EPO}(P) = \big(min(EPO(P)),\ max(EPO(P)),\ stdev(EPO(P)) \big)^T, \tag{2.6}$$

$$V_{EOH}(P) = \big(min(EOH(P)),\ max(EOH(P)),\ stdev(EOH(P)) \big)^T, \tag{2.7}$$

where, min, max and $stdev$ are functions that return minimum, maximum and standard deviation values of the elements of a vector, respectively. Figure 2.8 shows three images from the DTD dataset and their corresponding edge images. The first two images, from left to right, represent two instances of the same texture class (*'Banded'*) but presenting different orientations of bands (45° rotated bands). The third image is an instance taken from the *'Spiralled'* class. The ES matrices are extracted on three pixels from the three instances with a 4×8 distribution (4 orientations and 8 circles). The EPO and EOH features are then calculated from the ES matrix of each instance. Finally, statistical information from the two calculated features are extracted into the V_{EPO} and V_{EOH} vectors. The EPO and EOH features of the two images belonging to the same class include the same elements but shifted because of the rotation of bands between the two instances. This offset is no longer present in the V_{EPO} and V_{EOH} vectors thanks to the rotation invariance property of the min, max and $stdev$ functions when dealing with the same shuffled elements. However, the third image from the *'Spiralled'* class presents distinctive V_{EPO} and V_{EOH} vectors compared to the ones of the *'Banded'* class. The same illustration, for two images of the same instance with different scales and with an 8×8 size distribution, is presented in Figure 2.9. The extracted V_{EPO} and V_{EOH} on two pixels from the original image and the scaled one are almost the same.

Thus, the proposed LES descriptor is obtained by the concatenation of the two statistical vectors V_{EPO} and V_{EOH}, and it captures a highly discriminative information from the local texture, while being of low dimensionality. Indeed, the choice of the LES descriptor for the classification of texture images can be justified by the following points:

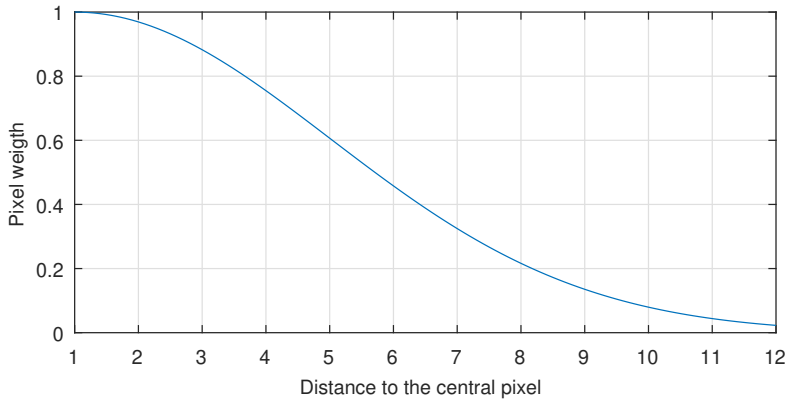


Figure 2.7: Vote weighting function for an edge pixel ($\sigma = 4$).

- **Rotation insensitivity:** *LES* is invariant to any rotation of $k.\theta$ degrees, where $k \in \mathbb{Z}$ and θ is the angular resolution of the used distribution. Rotation invariance could be generalized for any angle if a fairly fine angular resolution is used. This invariance comes from the rotation invariance nature of the statistics (*min*, *max* and *stdev*) that we used to define the descriptors.
- **Scale insensitivity:** The scale insensitivity of the V_{EPO} descriptor comes from the fact that each element of the vector is a statistic of the frequency of edge pixels across orientations that is not very affected by scaling. The insensitivity of V_{EOH} descriptor to the scaling is ensured by the weighting function. Indeed, the edge pixels closest to the center of the neighborhood are the most decisive in the vote and the determination of the orientation histogram.
- **Discrimination Power:** Both V_{EPO} and V_{EOH} are very discriminating when dealing with different textures. Indeed, these descriptors provide independent scale first-order statistics on edginess across orientations and edge pixels' orientations in a local neighborhood, which says a lot about the nature of a texture (*e.g.* if it is a line-like texture, regular texture, random texture, rough texture...).

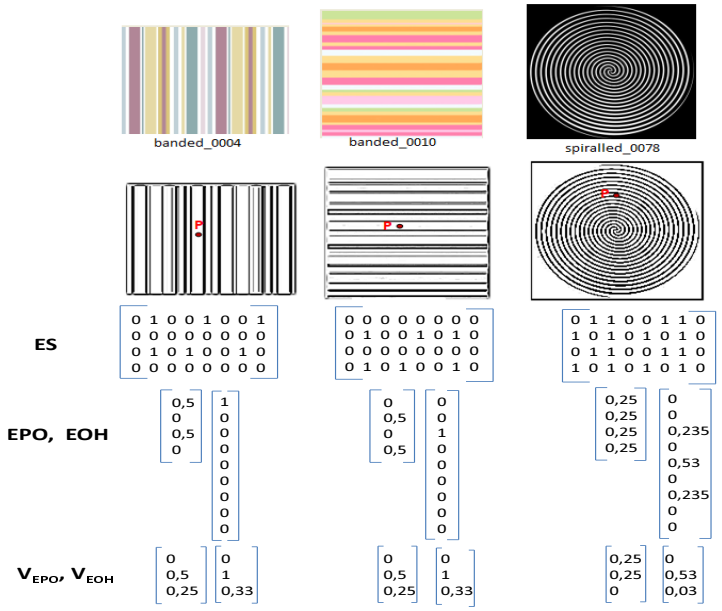


Figure 2.8: V_{EPO} and V_{EOH} for three texture instances, from two classes of the DTD dataset, using a distribution with $N = 4$ and $M = 8$ (4 orientations and 8 circles).

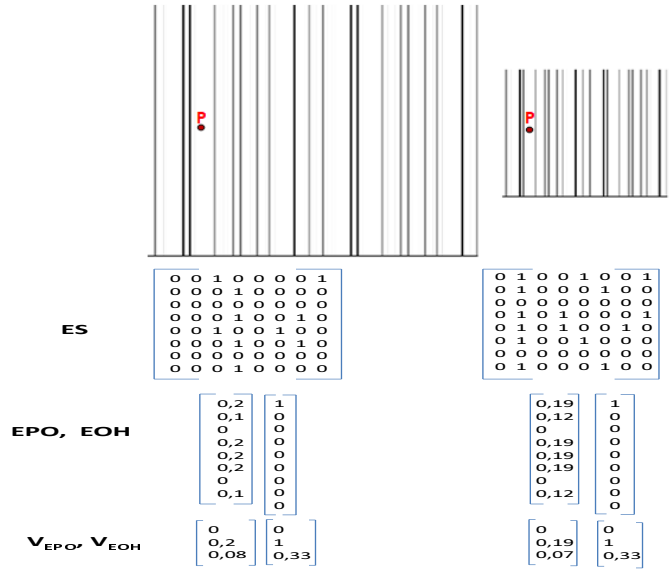


Figure 2.9: V_{EPO} and V_{EOH} of the same instance, from the DTD dataset, with two different scales using a distribution with $N = 8$ and $M = 8$ (8 orientations and 8 circles).

2.3.3 Global Texture Description and Classification

In this section, the overall process for global texture description and classification is detailed. Figure 4.2 shows the different steps of the classification process. First, training data are fed to a genetic programming process to generate a texture descriptor based on the proposed local edge signature (red arrows). Then, the genetically elected descriptor is used to generate a set of training features from the training data (blue arrows). Finally, the training features are used to learn a classifier (green arrows). The genetically learned descriptor generates the feature set for the testing data which will be fed to the trained classifier in order to label the unseen data. Furthermore, the main steps of the adaptation of genetic programming to any computer vision problem are individual representation, fitness function definition and variation operator design. Since the individual in our case is a descriptor, and the fitness function is calculated from the features it generates, the extraction of features is an important step. Thus, to understand how the suggested genetic process generates the global texture descriptor based on the proposed local edge signature, individual representation, feature extraction and fitness calculation steps should be described. Thereafter, since individuals evolved by genetic programming are descriptors in our case, and not classifiers, the same instances that were used to learn the global descriptor will be used to generate a set of training features. This set will be used to train a classifier. For the unseen data, the vector is generated using the evolved descriptor and fed to the trained classifier to label the input image. Indeed, we have tested four different types of classifiers to have more significant and unbiased results. The first classifier is the 1-NN (k-Nearest Neighbor (k-NN) with k set to 1) which is widely used in texture classification applications [21, 48, 68, 90, 58]. The second one is the SVM (Support Vector Machine) with a radial-basis kernel, $C = 1000$ and $\gamma = 50$ (for the same reasons mentioned in [37]). The third and fourth classifiers are the K^* and the NNGE (Non-Nested Generalized Exemplars) classifiers, while using the "1 closest neighbor".

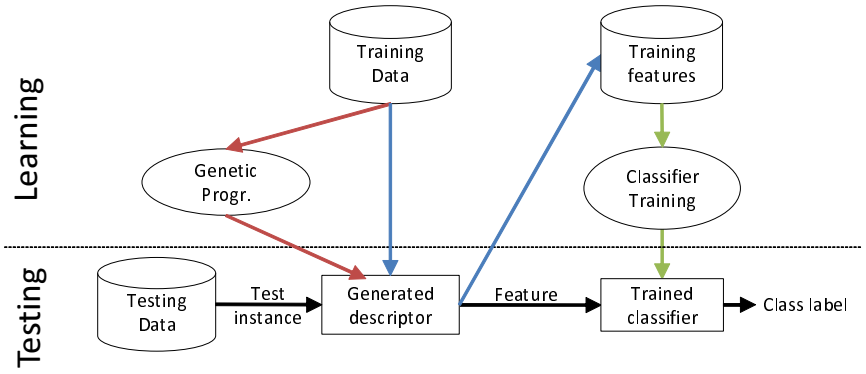


Figure 2.10: Overview of the proposed process for evolving a global descriptor and classifying texture.

A tree-based representation is used to randomly generate a population of individuals. An individual tree is made up of a root node, a number of internal nodes, and leaf nodes. An example of a Genetic Programming (GP) individual is depicted in Figure 4.6. The terminal set (leaf nodes) consists of nodes which are chosen randomly among the six *LES* vector elements previously detailed. LES_i designates the i^{th} element of the *LES* vector. The non-terminal set is composed of *root node*, *root children nodes* and *function nodes*. A *function node* is chosen randomly between a set of arithmetic operators $\{+, -, \times, /\}$, which have the same input and output type. An exception for the division operator, it returns zero if the denominator is zero. The *root node* is the *collector node* and is responsible for collecting the results from the children nodes. This node will be detailed later when explaining how the feature vector is generated. The root children are *sup* nodes. Their number will specify the size of the feature vector. The *sup* node (">" in Figure 4.6) is a binary operator that returns 1 if the left child is greater than the right child and 0 otherwise.

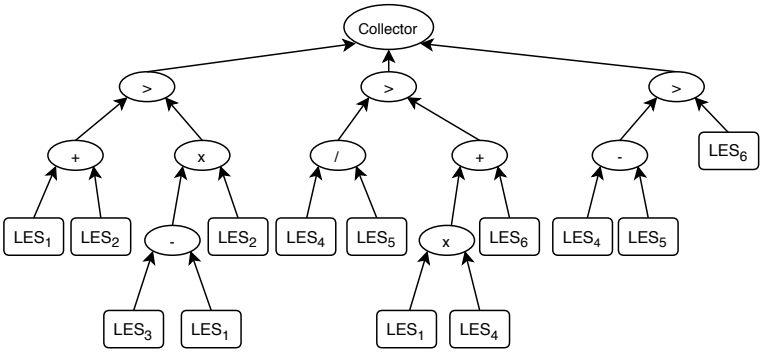


Figure 2.11: Example of an Individual tree structure for a 3-bit code descriptor.

Each individual (descriptor) is applied to a set of keypoints in the input image. For each keypoint, the *LES* vector is computed as described previously, and elements of the computed *LES* are used as terminal nodes of the descriptor. The non-terminal nodes are evaluated starting from the leaf nodes up to the root children by applying the corresponding operator to the child nodes. The collector node produces a binary code from the root children as shown in Figure 4.7. The length of the binary code is specified by the number of root children nodes. In the remaining of this paper, *code length* (cl) denotes the number of children of the root node of an individual. An individual with cl root children generates a $cl - bit$ binary code. This binary code is used to construct a 2^{cl} feature vector. The decimal equivalent of the generated binary code will indicate the bin of the feature vector for

which the vote will be allocated. Figure 4.7 illustrates how a tree-based individual, where the value of *code length is equal to* 3, transforms the *LES* feature, calculated in a given keypoint, to a vote within the global histogram feature. Each leaf node of the individual tree refers to an element of the *LES* feature, and the value of a parent node is defined in a bottom-up manner from the children nodes with the corresponding operator. Each root children returns a binary value (1 if the left child value is greater than the right one and 0 otherwise), and the collector node (root) collects the final binary code (= 101 in the example of Figure 4.7) and transforms it to its corresponding decimal number (= 5 in the example of Figure 4.7). The decimal number obtained represents the bin into the final histogram feature to which the vote will be allocated. The 3-*code length* descriptor in this example generates an 8-dimensional feature vector (= 2^3). The genetic encoding of the descriptors and the feature extraction being detailed, it remains to define the fitness function that will allow the GP algorithm to elect relevant descriptors.

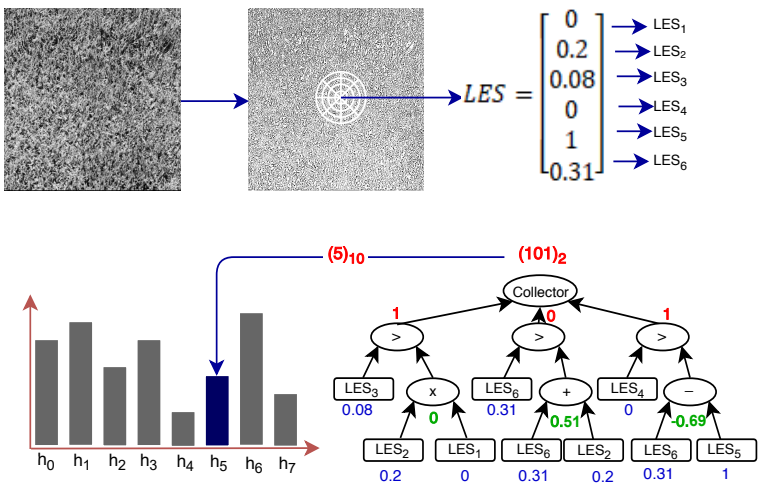


Figure 2.12: Feature vector generation: a 3-bit descriptor transforming the LES on a pixel to a vote in an 8-bit histogram feature.

In genetic programming-based classification, the rate of correctly classified instances is, generally, taken as a fitness measure. In our case, evolved individuals are descriptors and not classifiers. A good descriptor is the one that generates the best features to be fed to a classifier. To enhance the classification task, features must be close in the case of instances of the same class and very discriminating when it comes to different classes. The proposed fitness measure (4.8) takes into account the homogeneity of features inside each class and their discrimination power when dealing with different classes.

$$fitness = 1 - \frac{log(2)}{log(1 + e^{DC/HC})}, \tag{2.8}$$

where, *HC* (4.9) is the homogeneity coefficient that describes how strongly feature vectors of instances of the same class resemble each other, and *DC* (4.10) is the discrimination coefficient, which describes how strongly feature vectors of instances of different classes are distant from each other. Indeed, the *HC* (*resp.* *DC*) coefficient illustrates the average intra-class (*resp.* inter-class) similarity measure between training features. Both *HC* and *DC* coefficients range from 0 to 1. Good intra-class features homogeneity corresponds to *HC* values close to 0, and the discriminating power of a descriptor

grows for DC values nearby 1. Thus, the best individuals are those with larger DC/HC ratios (Figure 4.8).

$$HC = \frac{2}{m(m-1)n} \sum_{k} \sum_{i<j} s(X_i^{c_k}, X_j^{c_k}), \tag{2.9}$$

$$DC = \frac{2}{n(n-1)m^2} \sum_{k<l} \sum_{i,j} s(X_i^{c_k}, X_j^{c_l}), \tag{2.10}$$

where, n is the number of classes, m is the number of training instances per class, $X_j^{c_i}$ is the normalized feature vector of the j^{th} training instance of the class c_i ($i \in \{1, \ldots, n\}$), and $s(U,V)$ (4.11) is a measure of similarity, which ranges from 0 to 1, between two normalized feature vectors of the same dimension p.

$$s(U,V) = \frac{1}{p} \sum_{i=1}^{p} \frac{|u_i - v_i|}{u_i + v_i}. \tag{2.11}$$

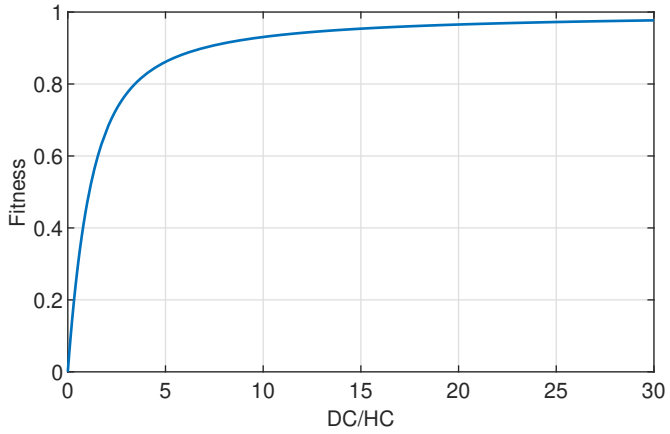


Figure 2.13: Fitness measure as a function of the DC/HC ratio.

2.3.4 Results

In this section, the proposed texture classification method is tested and compared to nine relevant texture classification methods. The experiments have been executed on a PC with Windows 10 as an operating system with Intel®$Core^{TM}$ i7-7500U CPU @ 2.70 GHz and 8G of memory. It is worth noting that the measured time in all our experiments is the CPU time and not the wall-clock time. The classification accuracy is used as a metric to evaluate the performance of each individual to differentiate between instances of various classes. Furthermore, two validation protocols are used for the experimental results. The first protocol (*Prot.I*) is intended to demonstrate that the proposed method works well with a limited number of training data (5 trainig instances) when it comes to datasets with reduced number of classes (10 to 25 classes). In *Prot.I*, the images of each dataset are divided into two disjoint subsets. The first sub-set is constructed with 50% of the samples of each class, which are randomly chosen. A number of samples (5 for the comparison results), randomly chosen from this sub-set, are used for the training. The second sub-set, constructed with the other 50% of the samples of each class, is used for the test. The reported results will be the average accuracy over 100 executions of each method and for each texture dataset for the non-genetic-based methods. For the genetic-based methods, the descriptors are learned according to the genetic parameters specified

by the compared methods and the reported results will follow the same protocol as non-genetic-based methods. For each dataset, the method with the best performance for each classifier is made bold and the worst one is marked by the '$*$' symbol. The second protocol (*Prot.II*), which is the same protocol reported in [63], is a 10-fold cross-validation scheme reporting the median accuracy of 11 independent runs. *Prot.II* is used to investigate the proposed method performance on datasets with large number of classes (68 and 111 classes), while running the evolution process using a larger training set. To avoid overfitting and to keep a reasonable computational time, a subset of the training set is used and it is randomly changed through generations of the GP process.

The proposed classification method is evaluated along with six challenging image datasets for texture classification. These standard datasets vary in the number of classes, the size of instances (form 128×128 pixels to 640×480 pixels) and the various texture materials (foliage classification, brick, wall...). Moreover, as summarized in Table 4.3, these datasets vary in illumination, scale, point of view, rotation, number of classes and number of instances per class. The datasets mentioned in Table 4.3 are primarily sorted in ascending order based on the complexity of changes (illumination, viewpoint, rotation, scale) within the instances and then sorted by the number of classes. Some of the used datasets present color information in their instances. In this work, we consider only the grayscale level of these images, which is obtained from pixel luminance [63, 4].

				Changes				**Size**	
Datasets	**# Classes**	**# Instances**	**Prot.**	**Illum.**	**View**	**Rotat.**	**Scale**	**W**	**H**
Brodatz Prot.I/Prot.II	15/111	375/2775	I/II	×	×	×	×	128	128
Outex_TC_0000	24	480	I	✓	×	✓	×	128	128
Outex_TC_0013	68	1360	II	✓	×	✓	×	128	128
KTH-TIPS	10	810	I	✓	✓	×	✓	200	200
KTH-TIPS2b	11	4752	I	✓	✓	×	✓	200	200
UIUCTex	25	1000	I&II	✓	✓	✓	✓	640	480

Table 2.3: A summary of the used benchmark image classification datasets.

Comparison results on five datasets are presented in Table 2.4. Using a 1-NN classifier, the proposed method has achieved the best accuracies on average and outperformed all other methods on the five used datasets. The best accuracy was obtained on the Outex_TC_0000 dataset and has reached 95.6%. Worst performance for the proposed classification method was carried out with the NNGE classifier (= 75.2% for the UIUCTex dataset). However, except for the case of a 1-NN classifier, $GP - criptor$ and $GP - criptor^{ri}$ outperformed the proposed method for the Brodatz and Outex_TC_0000 datasets, while using the remaining classifiers. This can be explained by the fact that these two datasets do not include scale changes. Furthermore, results of the proposed method on the very challenging UIUCTex dataset reached over 84% with both NNGE and 1-NN classifiers with only 5 training instances per class (*Prot.I*). This could reveal the ability of the proposed method to handle all types of changes and deformations involved in the UIUCTex dataset. For the KTH-TIPS and KTH-TIPS2b datasets, the proposed GTS method outperformed all the other methods for all the classifiers, what confirms the scale insensitivity of the GTS as argued earlier in this paper. In addition, the suggested method achieved the best performance on the Brodatz dataset (15 classes), with the 1-NN classifier, compared to all the other methods. But, it was slightly outperformed by $GP - criptor^{ri}$, with the SVM classifier, and $GP - criptor$, with K* and NNGE classifiers. These two methods also generate automatic descriptors, what supports the hypothesis of the superiority of automatic methods over expert ones. In all other cases, the quality of the features generated by

	LBP^{u2}	LBP^{riu2}	LBC	$CLBC$	$GLCM$	DIF	$GPcrt$	$GPcrt^{ri}$	$ARCS$	GTS
					1-NN					
Brodatz (15 classes)	72.2	78.3	75.3	78.5	68.3	38.2*	91.8	91.6	88.9	**93.7**
Outex_TC_0000	68.6	71.5	70.4	72.3	59.8	35.8*	62.4	90.2	92.3	**95.6**
KTH-TIPS	47.7	50.8	59.9	58.2	41.6	23.4*	51.3	75.5	88.4	**91.2**
KTH-TIPS2b	42.1*	49.3	60.3	55.9	41.8	24.5	52.7	70.8	91.1	**94.3**
UIUCTex	29.4	42.7	49.8	45.1	31.5	12.9*	47.3	69.2	77.9	**84.3**
					SVM					
Brodatz (15 classes)	71.2	70.8	74.9	71.2	52.8	32.1*	90.1	**85.6**	76.4	84.4
Outex_TC_0000	65.3	72.6	71.6	65.3	51.2	29.8*	64.3	84.7	80.6	**85.6**
KTH-TIPS	43.6	47.1	58.6	52.7	39.5	18.6*	41.2	68.9	71.6	**83.2**
KTH-TIPS2b	41.2	52.0	55.7	52.9	42.9	19.2*	39.5	66.1	78.8	**83.7**
UIUCTex	25.3	41.9	48.0	41.9	29.3	11.0*	40.4	61.5	71.3	**79.6**
					K*					
Brodatz (15 classes)	68.1	71.2	75.5	80.3	70.6	38.5*	**91.8**	89.9	84.5	89.2
Outex_TC_0000	64.2	68.0	72.8	79.5	60.9	36.9*	62.3	**88.3**	79.6	84.3
KTH-TIPS	40.3	56.4	59.7	57.6	45.3	30.6*	51.1	70.5	75.4	**88.5**
KTH-TIPS2b	38.4	54.3	56.2	52.4	47.2	29.8*	47.1	71.4	78.6	**82.9**
UIUCTex	27.7	41.8	43.8	47.7	38.7	13.8*	33.7	66.8	76.3	**84.6**
					NNGE					
Brodatz (15 classes)	70.5	69.7	73.8	79.8	64.6	40.3*	**88.4**	80.3	72.2	80.5
Outex_TC_0000	66.4	68.2	70.3	76.8	60.2	38.6*	61.2	**79.9**	74.6	78.5
KTH-TIPS	32.9*	53.8	56.7	56.4	52.4	35.0	44.3	61.8	74.0	**80.2**
KTH-TIPS2b	42.3	50.6	54.8	51.9	53.0	29.8*	39.5	59.7	71.3	**78.4**
UIUCTex	30.2	42.5	42.6	51.3	39.4	17.6*	31.6	55.8	72.9	**75.2**

Table 2.4: Comparison of the proposed method (*GTS*), against relevant state-of-the-art methods, on five datasets while testing four different classifiers with 8×10 distribution using *Prot.I*.

the *GTS* either significantly improved the performance or achieved a comparable performance (best or in the top-three ranked accuracies) to that of the other methods. This proves that the automatic aggregation of local features by genetic programming can give better feature than expert designed ones. Although the performance of the *GTS* method has slightly degraded with the SVM, K*, and NNGE classifiers, in comparison to the 1-NN classifier, it shows a significant improvement in its performance compared to the other image descriptors. It is worth noting that, for this experiment, the training was performed with 5 random instances per class for the proposed method. This proves that it can even deal with classification problems with few labelled data. Nevertheless, all the datasets used in this experiment have relative small number of classes.

Table 2.5 summarizes the comparison results, on three datasets, of the proposed method against five relevant methods from the state-of-the-art. For the Outext_TC_00013 dataset, with 68 classes, the proposed method outperformed all the others by scoring an accuracy of 90.2%, which is a very encouraging performance given the difficulty of this dataset. For the Brodatz dataset, the suggested method records the second performance (96.9%) after the *Lower and Upper Volume*. The performance for this dataset has even been improved compared to the experiments' run with *Prot.I*. Knowing that 15 and 111 classes from the Brodatz dataset were used with *Prot.I* and *Prot.II*, respectively, we can notice that the *GTS* method can deal with large number of classes if we increase the size of training set (from 5 to 22). The experiments on the UIUCTex dataset were run using the same number of classes

(25 classes) as in *Prot.I*. The fact that the number of training instances was increased while keeping a relatively small number of classes can lead to overfitting and performance degradation on unseen data. The proposed method has overcame this problem and the performance was even improved from 84.3% to 91.1%. That is because we are using only a subset of the training data for directing the evolution and randomly changing this subset at every generation. Some works have demonstrated that the smaller this subset is, the less overfitting occurs [45]. This evolution strategy makes it possible to have a descriptor that is efficient in the whole training set while avoiding overfitting and keeping a reasonable computational time. To summarize, the proposed method works very well with a small number of labelled samples in datasets with relatively small number of classes. When it comes to datasets with large number of classes, increasing the size of the training set and dividing it randomly through the generations of the evolution process, have given even more accurate classification results while keeping almost the same computational time.

			Datasets		
Method	**Reference**	**Validation**	**Outext13**	**Brodatz**	**UIUC**
$\vec{\Psi}_{19,39}$	[63]	10-fold	89.7	95.2	-
$\vec{\Theta}(39)_{3,5,7}$	[63]	10-fold	88.8	94.1	-
L&UVolume	[4]	leave-one-out	85.5	**99.0**	76.3
FFTFE	[144]	10-fold	89.6	-	90.8
ARCS – LBP	[100]	10-fold	85.7	92.8	81.2
GTS	Proposed	10-fold	**90.2**	96.9	**91.1**

Table 2.5: Comparison with relevant state-of-the-art methods using *Prot.I*

2.3.5 Conclusion

In this chapter, first, a GP method has been proposed to automatically achieve patch detection, feature extraction and binary classification of texture images under different effects. This method ensures a certain robustness when dealing with scale, pose, illumination and rotation variations through a set of functions inspired by HOG and LBP descriptors. Differently to existing methods, the proposed GP method does not require human intervention and uses only a limited number of instances per class to evolve a classifier. It is suitable for problems where only a small set of labeled samples is available. To examine its performance, extensive experiments have been conducted on six challenging texture collections of varying difficulty with different effects and degrees of rotation. The proposed method has been also compared to several relevant methods from the state-of-the-art, and the results show that HL-GP works very well for binary texture classification. In fact, the combination of HOG and LBP proved very effective for the extraction of high level features using genetic programming. Second, a robust method of automatic generation of texture descriptors through genetic programming is proposed. Texture is described locally using a new descriptor called Local Edge Signature (LES). We have reported in this descriptor, in a manner faithful to the human perception of the texture, statistics related to the arrangement of edges in a local region. The nature of the local neighborhood, as well as the statistics used in the proposed method, make the description of the texture as insensitive as possible to changes of scale and rotation. Moreover, we adapted a genetic programming technique to automatically generate a global descriptor from a set of keypoints. The proposed texture classification method has been tested on the most challenging datasets. The reported performances were among the best results recently found for texture classification. Improvements can still be made to the suggested

method. For instance, the parameters of the local distribution can be encoded in individuals. This may lead to increase the size of the initial generation in order to ensure more diversity. In this case, the use of GPUs is conceivable to reduce the learning time.

Chapter 3

Facial expression recognition

The main results presented in this chapter have been published in the following international journals: Multimedia Tools and Applications (MTAP 2019) [9] and Applied Soft Computing (ASC 2021) [38] in addition to the main conferences: International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS 2017) [7] and IEEE/ACS International Conference on Computer Systems and Applications (AICCSA 2021) [16].

3.1 Introduction

Human Facial Analysis (HFA) is a major field of research in computer vision and pattern recognition. In fact, the growing interest in the analysis of human faces comes from its ability to reveal the demographic information (gender, age, ethnicity...) [50] and the person's identity [117]. Moreover, HFA is considered as an important emotional and awareness communication channel that reflects some of our cognitive activities and well-being [116]. The pioneering study conducted in [30] proved the universality of six facial expressions (happiness (HA), anger (AN), sadness (SA), fear (FE), disgust (DI) and surprise (SU)). Indeed, peoples from different cultures show the same facial expressions for the same feelings. The strong acceptance of this affirmation in psychology is motivating researchers in computer vision and affective computing to develop automated systems for the recognition of emotional states and human affects from facial expressions. Historically, facial features analysis started with 2D still images, and many applications were implemented, such as Facial Expression Recognition (FER) under rigorously constrained conditions. Recognition of human emotions has long been the subject of active research area. A wide range of human interaction applications have to decipher the facial emotional state. Unlike other non-verbal gesture, the emotional state of the face can be relied to several expressions. Most research has focused on posed facial expressions and reached high level of efficiency recognizing human emotions [8, 9]. However, some posed expressions are still very hard to discriminate such as *disgust* and *sadness* emotions. Quite fewer works have done advances interpreting spontaneous facial emotions. There are several factors affecting the precision of facial expression recognition (FER) systems on spontaneous or posed expressions, including prominent facial feature selection, feature fusion and classifier design. Since, FER applications have to deal with natural emotions, our first goal was to develop a system that can achieve accurate recognition rates on posed as well as on spontaneous facial expressions.

Extracting efficient facial features is crucial towards facial emotion recognition. Commonly, two types of features are used to discriminate facial emotions: geometric and appearance features. Geometric features give clues about shape and position of face components, whereas appearance based features contain information about the furrows, bulges, wrinkles, etc. Appearance features contain micro-

Figure 3.1: Two faces displaying *disgust* emotional state that were miss-classified as *sadness* using geometric features according to a study in [134].

patterns which provide important information about the facial expressions. But one major drawback with them is the difficulty to generalize across different persons. Although geometric features are noise sensitive and difficult to extract, they proved to be sufficient to give accurate facial expression recognition results [146]. Moreover, He et al. [55] demonstrated that geometric features are more effective than appearance ones in most cases. However, geometric-based FER methods still have difficulties discriminating some expressions. As an illustration, Figure 3.1 shows two faces displaying *disgust* emotional state that were miss-classified as *sadness* expressions using geometric based features and correctly recognized using local binary patterns (LBP) according to a study presented in [134]. Indeed, micro-patterns, captured by LBP features, are able to offset the weakness of geometric based features by capturing micro-variations caused by wrinkles, which are useful to separate the *disgust* and *sadness* emotional states. Therefore, facial geometry distortions, given by geometrical features, are complementary with textural information captured by appearance ones. In other words, considering geometrical and appearance feature fusion can be an interesting way to design more discriminative features to deal with FER challenges.

Feature fusion based methods face the problem of high dimensionality which can affect the quality of the facial emotion recognition. Indeed, dealing with large number of features can increase the computational time and overwhelm classifiers with unnecessary or redundant information. In this case, a rigorous feature selection step is necessary. To carry out selection of a good feature subset, several factors must be considered. First, feature selection cannot be performed in the same way for spontaneous and posed expressions. Indeed, spontaneous facial muscle movements have been proven significantly different from deliberate ones. According to Ekman [32], zygomatic major is the only interacted muscle in posed smiles. However, muscles around the eyes (i.e. orbicularis oculi) are also contracted during genuine smiles. Moreover, Namba et al. [106] studied the difference in action units (AUs) [31] between involuntary and real emotions. For instance, the three most commonly seen AUs for genuine *disgust* are squinting eyes and raising the upper lip. On the other side, glare and raising the chin are often spotted in posed *disgust*. The eyebrow raiser, the lips part, the jaw drop and the upper lid raiser are hardly observed in genuine expressions comparing to acted ones as specified in [106]. In other words, for the same emotional state, spontaneous AUs differ from posed ones. Second, AUs discriminating between expressions may change from one couple of expressions to another even within the same expression category (spontaneous or posed). Therefore, performing a static selection method and choosing relatively the best features subset to implement the prediction for all the expressions may not be efficient. In fact, choosing the average does not always mean choosing the best. Although the selected subset has shown good results in most expressions, it may perform poorly in

others. Thus, for better training and emotion detection, choosing the right and effective features is crucial as irrelevant and noisy features may mislead and negatively affect the recognition system.

In this chapter we present our contributions within the frameworks of 2D and 3D/4D FER. In the field of 2D facial expression recognition, two main challenges facing research were raised. First, is it possible to fuse hybrid features (geometric, texture, etc.), within the framework of facial expression recognition, without information redundancy? Second, how to design a feature selection mechanism to solve the problem of expressions that are commonly miss-classified by FER systems? The section 3.2, presents the work based on the paper [38], which explores soft computing techniques, such as genetic programming, to enhance the way features are selected and fused for more accurate 2D facial expression recognition. In the section 3.3, our contributions within the framework of 3D/4D FER, mainly based on the papers [8, 9] are presented.

3.2 2D facial expression recognition

3.2.1 Motivation, contributions and overview

Feature selection and fusion can be considered as a search-based optimization problem since they rely on searching an optimal subset of features to perform an accurate classification. Evolutionary computation (EA) is a family of biologically inspired optimization algorithms based on trial and error problem solvers that can be applied in this case. Evolutionary algorithms (EA) are a subset of EC that involves techniques implementing biological evolution-inspired mechanisms. Genetic algorithm (GA) belongs to the larger class of EA. It is a metaheuristic based on the theory of evolution by natural selection. Genetic algorithms are randomized search algorithms that produce high quality solutions for search and optimization problems. They operate according to the rule of survival of the fittest and rely on biologically inspired operators such as crossover, mutation and selection. Genetic programming (GP) is an evolutionary approach that extends genetic algorithms to evolve computer programs. Like other evolutionary techniques, it starts from a population of random programs and fits them for a particular task by defining a goal in the form of an evaluation criterion also referred to as fitness function. This fitness function is thereafter used in order to evolve a population of candidate solutions, called individuals, based on the process of Darwinian evolution. GP works using an iterative fashion, where each iteration involves the probabilistic selection of the fittest programs and their variations using a set of genetic operators: crossover and mutation. The crossover operator works by swapping random parts of selected parent programs to produce new and different offspring that become part of the next generation. Mutation operator involves random substitution of a part of a program. Typically, individuals of the new generation are on average more fit than those of the previous one. The iterative process of evolving programs stops when one or more programs reach a preset fitness level. A premature convergence to local optimum as well as overfitting may occur if the initial population size, the maximum number of iterations and other genetic parameters are not well chosen. In the framework of 2D FER, we propose a GP-based framework for hybrid feature selection and fusion for facial expression recognition. A subset of features is selected and fused differently for each pair of emotion classes. A three-layer tree-based representation of a GP-based program is defined for each pair of expressions to perform feature selection, feature fusion and binary classification. The multi-class recognition is performed using the binary GP-based programs. The proposed framework is tested on the combination of three geometric and appearance features but can be generalized to any features by simply modifying the set of terminal functions in the selection layer. The main contributions we present in the framework of 2D FER are threefold:

- We propose a genetic programming (GP) based framework for posed and spontaneous facial expression recognition allowing to combine hybrid facial features, then we test it on the fusion of geometric and appearance features.
- The feature selection and fusion, in this work, are performed in a binary way: the most prominent features are selected and fused differently for each pair of expressions.
- To the best of our knowledge, we are the first to propose a genetic programming based classifiers for facial emotion recognition incorporating simultaneous feature selection and fusion within the evolutionary learning process.

The general flowchart of the suggested method is illustrated in Figure 3.2. Indeed, a face detection step followed by a feature extraction step are performed. In the extraction step, geometric and texture features are extracted and fed to genetically evolved programs. Indeed, a binary genetic program learns to select the most discriminating features and to fuse them specifically for each pair of expression classes. There are as many learned binary classifiers as there are pairs of expression classes. The predicted expression for the input image is captured by performing a unique tournament elimination between all the classes using the learned binary programs.

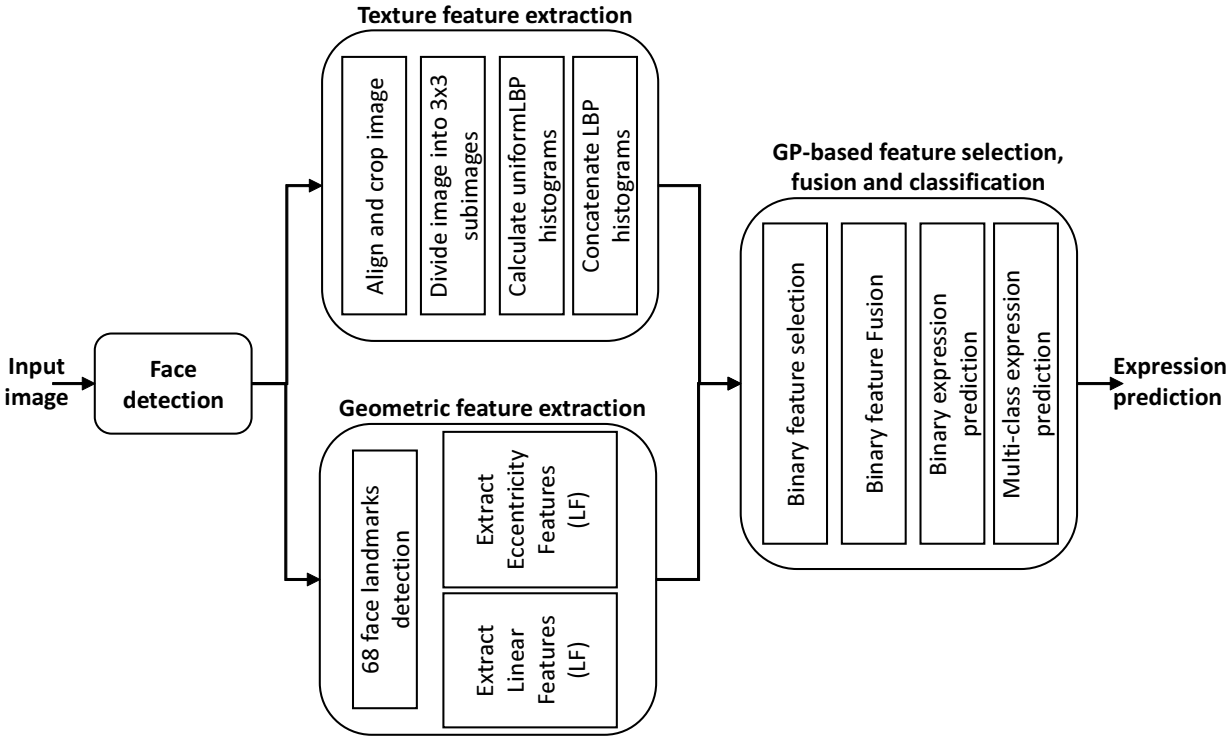


Figure 3.2: Flowchart of the proposed method for 2D FER.

3.2.2 Face detection and feature extraction

Detecting the human face in the input image is the initial step for the proposed method. The face detection algorithm uses histograms of oriented gradients (HOG) [23] with support vector machines (SVM) [11]. Afterward, landmark detection and face alignment algorithm based on a set of regression trees as described in [73] and trained in the iBug 300-W dataset [131], are performed to locate 68 facial keypoints and to align the input faces. The indexes of the 68 face landmarks are shown on Figure 3.3.

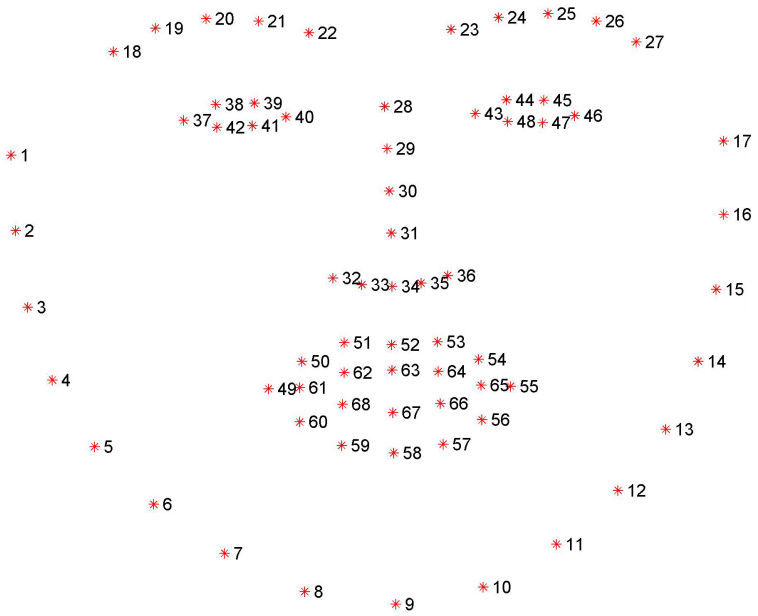


Figure 3.3: The set of 68 facial keypoints detected.

The geometric features present a general description of facial shape, curvatures, location and distance between facial components such as mouth, eyes, nose and eyebrows. These geometric properties provide important information about facial deformation during the emotional display. Thus, after extracting the 68 keypoints, we investigate the geometrical relation between landmark positions. However, as explained in the introduction section, geometric features can be complemented by appearance ones to deal with challenging expressions. In this work, we consider the fusion of geometric and appearance features.

3.2.3 Geometric Feature Extraction

(a) (b)

Figure 3.4: Two emotional states with approximately the same facial gestures: (a) anger, (b) fear.

The geometric features present a general description of facial shape, curvatures, location and distance between facial components such as mouth, eyes, nose and eyebrows. These geometric properties provide important information about facial deformation during the emotional display. Employing only the distance or angles between facial components may not be very effective in some cases. As an illustration, Figure 3.4.a and Figure 3.4.b present a male subject displaying anger and fear, respectively. As it can be seen, the face gestures in the two emotional states are approximately the same. The human eye can not even perceive the difference between these two emotional states. Depending only on the

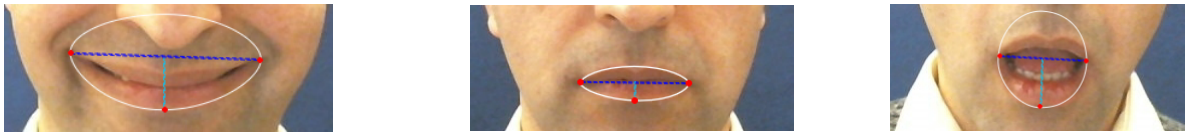
distance between facial features, in this case, may not be quite practical to correctly discriminate between the two emotions. Furthermore, while investigating the shape of facial components, in this instance, the eyes, play a key role in discriminating the two expressions. Thus, eccentricity is one of the well-known measures for providing shape description. Therefore, taking into consideration the elliptical shape of the face components in addition to the distances between landmarks can provide a better description of the facial behavior during the emotional state. Thus, after extracting the 68 keypoints, we investigate the geometrical relation between landmark positions. For better expression representation, we consider two types of geometrical features: linear and eccentricity features. For the linear features (LF), Euclidean distance between all pairs of landmarks, as given in Equation 3.1 are commonly used to capture the facial activities during the emotion deliberation where x_1 and y_1 present the (x,y) coordinates of the first landmark and x_2 and y_2 are the coordinates of the second one.

$$d(\mathbf{X}, \mathbf{Y}) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}, \tag{3.1}$$

Since 68 keypoints are extracted in this work, 2278 $(68(68-1)/2)$ different linear features can be calculated. For more robustness against the human face physical variation, all the distance are normalized as shown in Equation (3.2), where $Ncoef$ is a normalization coefficient. The Euclidean distance between the right corner of the left eye (L_{43}) and the left corner of the right eye (L_{40}) was chosen to calculate $Ncoef$ due to its stability during the different facial deformations.

$$\begin{cases} \mathbf{Nd}(\mathbf{X_i}, \mathbf{Y_i}) = \dfrac{d(\mathbf{X_i}, \mathbf{Y_i})}{Ncoef} \\ \mathbf{Ncoef} = d(L_{43}, L_{40}) \end{cases} \tag{3.2}$$

The eccentricity is a geometric term defining the ovalness level of an ellipse. In a more formal way, the eccentricity (e) is the ratio of the ellipse foci (c) to its semi-major axe (a). In fact, if the eccentricity is close to zero then the ellipse has a circular form. However, if it is close to one, then the ovalness of ellipse is high. For this work, the eccentricity of the mouth, eyes and eyebrows, as presented in Table 3.1, are considered. Taking into consideration the elliptical shape of these facial components, the eccentricity features may provide important information about their geometrical shape modification throughout the emotional state. For instance, during the display of surprise, the mouth is generally wide open presenting more of a circular form. However, the mouth is more long and skinny when the person is smiling displaying an ellipse-like shape. The same difference can be denoted for the eyes during the *surprise* and the *disgust* or *sadness* reaction. Figure 3.5 presents an illustration of the mouth deformation during *happiness*, *surprise* and *neutral* emotion display. The eight facial ellipses used in this work are presented in Figure 3.6.



(a) (b) (c)

Figure 3.5: An illustration of the lower lip elliptical shape during the display of (a) *happiness* (b) *neutral* and (c) *surprise*.

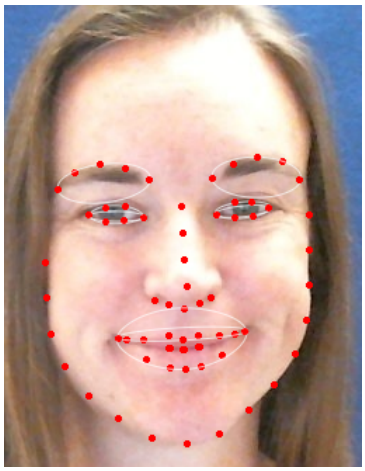


Figure 3.6: The representation of the eight facial ellipses.

Feature	Component	Landmark indexes
Ec1	Upper mouth	(49,52,55)
Ec2	Lower mouth	(49,58,55)
Ec3	Upper left eye	(37,39,40)
Ec4	Lower left eye	(37,41,40)
Ec5	Upper right eye	(43,44,46)
Ec6	Lower right eye	(43,48,46)
Ec7	Left eyebrown	(18,20,22)
Ec8	Right eyebrown	(23,25,27)

Table 3.1: The ellipses used to calculate the eccentricity features

3.2.4 Texture feature extraction

For computational and simplicity reasons, the local binary pattern (LBP) histograms are selected to be used as textural features. The LBP operator, introduced by Ojala [111], is one of the most widely used descriptors for feature detection and extraction. It locates keypoints within an image and generates a histogram that corresponds to their distribution. The LBP operator scans the pixels using a sliding window and generates a binary code based on the differences between the central pixel and its equidistant circular neighbors. The distance is defined by the radius parameter r and the number of neighbors is denoted by the pixel parameter p. The representation of the LBP operator is defined as follows (3.3):

$$LBP_{p,r} = \sum_{i=0}^{p-1} \mathbb{1}_{\mathbb{R}^+}(I(x_i,\ y_i) - I(x_c,\ y_c)).2^i, \tag{3.3}$$

where, $\mathbb{1}_S$ denotes the characteristic function of a subset S, x_c and y_c are the coordinates of the central pixel, x_i and y_i (3.4) are the coordinates of its i^{th} neighbor within the input image I.

$$\begin{cases} x_i = x_c + r.cos(2\pi i/p) \\ y_i = y_c - r.sin(2\pi i/p) \end{cases} \tag{3.4}$$

Figure 3.7: The original face image and the the cropped image divided into 9 sub-images of size 40×40.

The uniform local binary pattern operator $LBP^{u2}_{p,r}$ is used in this work to extract texture based feature. Indeed, the basic $LBP_{p,r}$ operator produces a 2^p different output values corresponding to the different binary patterns that can be formed by the p pixels. Experiments carried out in [111], have shown that some bins contain more information than others and it is possible to use only a subset of the 2^p bins to describe texture information. A LBP pattern is called uniform if it contains at most two bitwise transitions considering the circular shape of the binary string. The uniform patterns account for about 90% of the patterns for the $LBP_{8,1}$ according to the experiments performed in the original work. The $LBP_{p,r}$ variant accumulates all the non-uniform pattern in a single bin. For our case, the $LBP_{8,1}$ is used, which gives 58 uniform patterns and a total number of 59 bins considering the non-uniform bin. The histogram of a LBP labelled image $I_{LBP}(x, y)$ is given by equation 3.5.

$$H_i = \sum_{x,y} E(I_{LBP}(x, y) = i), \quad i = 0, \ldots, n-1 \tag{3.5}$$

Where n is the number of different LBP pattern labels and the function E is defined by $E(A) = 1$ if A is true and 0 otherwise.

The LBP histogram contains information about the distribution of the local micro-patterns, so it can be used to statistically describe facial characteristics. To represent texture in different face locations, input face images have been aligned and cropped to a uniform spatial resolution of 120×120 pixels and then divided into 9 equal-sized sub-images of 40×40 pixels each, as depicted in Figure 3.7. Subsequently, uniform pattern LBP histogram bins have been calculated for each sub-image separately and concatenated to form a global histogram for the entire face. Therefore, each image is represented by a histogram of 59×9 (=531) bins.

3.2.5 Learning binary programs by genetic programming

In this section, the overall process for genetically learning a binary programs for a specific pair of expression classes is detailed. Figure 4.2, shows the different steps for generating a GP program that performs feature selection and fusion to separate two facial expressions labeled A and B. The goal is to evolve a program that can predict the most likely emotion between A and B for a given input face image. First, the GP process randomly generates a population of tree-based programs (*i.e.* binary classifiers). Then, this population is evolved using classical genetic operators (crossover, mutation...). The generated classifiers are evaluated with a fitness function based on the training data involving instances from both classes A and B. Finally, the genetically elected classifier is used on unseen faces to predict the most likely facial expression between A and B. To understand how the suggested binary program performs binary emotion recognition, we have to focus on its structure.

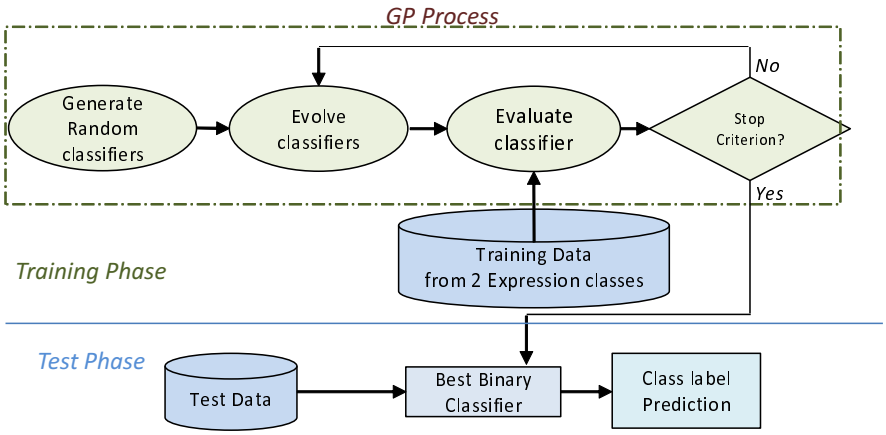


Figure 3.8: Overview of the proposed process for evolving a binary program.

Indeed, the proposed classifier has a three-layer tree-based structure. Each tree is made up of a root node (upper layer), a number of internal nodes (mid-layer), and leaf nodes (lower layer). A simplified representation of the proposed binary classifier is depicted in Figure 4.6. The binary classifier is evaluated in a bottom-up manner producing a single output value. The lower layer performs feature selection. This layer is composed of a set of three terminal functions, corresponding to the three extracted features, including the $Ec(i)$, $Nd(n,m)$ and $Bin(j)$ functions. The $Ec(i)$ function returns the value of the eccentricity value Eci as shown in table 3.1 ($i \in \{1, \ldots, 8\}$), the $Nd(n,m)$ calculates the normalized distance between two distinct landmarks n and m ($n,m \in \{1, \ldots, 68\}$) (Figure 3.3) as given in Equation 3.2, however the $Bin(j)$ function returns the value of the j^{th} bin ($j \in \{0, \ldots, 530\}$) of the LBP histogram described in Section 3.2.4. The parameters of these functions are randomly chosen among the allowed values while constructing the classifier. The features, selected in the lower layer, are fused in the mid-layer using operator nodes which are chosen randomly among a set of arithmetic operators $\{Add, Sub, Mul, Div\}$. These nodes have the same input and output type and perform classical arithmetic operations. An exception for the division operator, it returns zero if the denominator is zero. The upper layer performs the binary classification task based on the final value returned by the root node which is also an arithmetic node. This layer assigns an emotion label to the input face after comparing the value obtained with a predefined threshold (0 in our case). Genetic programming works in an iterative fashion, where at each iteration, a population of binary programs is evolved. Individuals of the initial population are generated by randomly selecting arithmetic operators and then applying them on the selected features. These features are obtained by randomly selecting their corresponding functions and their parameters from the terminal set. For each individual in a population, a fitness value is then calculated to evaluate its performance. In this study, the fitness is simply the rate of the correctly classified emotions among the set of the training facial expressions from A and B classes. Algorithm 2 presents the iteration sequences to generate a binary classifier program to discriminate between a couple of expressions A and B using genetic programming.

Algorithm 1: Generate binary program using GP

Input:
$(I_1^A,\ldots,I_m^A)$: Images from Expression A
$(I_1^B,\ldots,I_m^B)$: Images from Expression B
$MaxDepth$: Maximum program tree depth
N: Size of initial population
Result:
P_{best}: Binary classifier program for expressions A and B

```
1  begin
2      for i = 1 to N do
3          P_i ⟵ GenerateRandomProgramTree(MaxDepth)
4      Current_pop⟵ {P_1,...,P_N}
5      BestFitness⟵0
6      while stop-criterion is not reached do
7          for P_i ∈ Current_pop do
8              Fitness⟵0
9              for j = 1 to m do
10                 if EvaluateTree(P_i,I_j^A)>0 then
11                     Fitness⟵Fitness+1
12
13                 if EvaluateTree(P_i,I_j^B)<0 then
14                     Fitness⟵Fitness+1
15
16             if Fitness>BestFitness then
17                 P_best ⟵ P_i
18         Current_pop⟵evolve(Current_pop)
19     return(P_best)
```

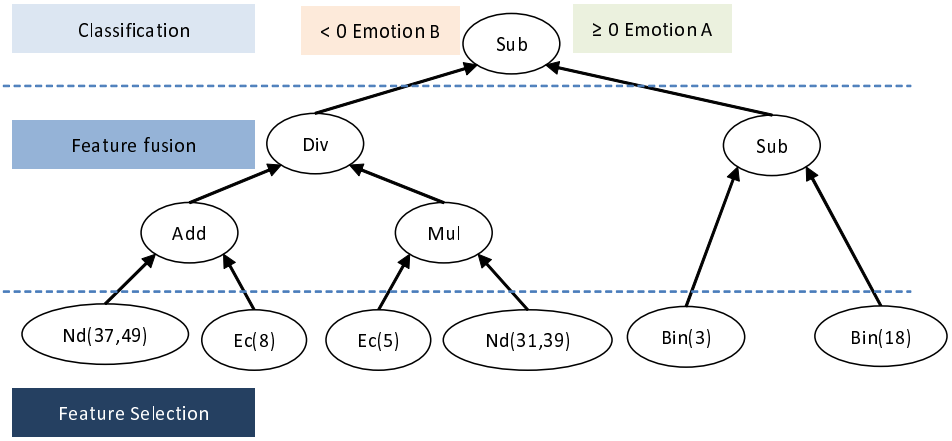


Figure 3.9: Simplified tree representation of a binary program.

Furthermore, the parameter settings of the genetic process for evolving the binary emotion classifier is summarized in Table 4.2. In fact, the ramped half-and-half method is used to generate the initial population, such that the population size is set to 200 individuals. The tournament selection strategy with a tournament of a size 7 is used to maintain the population diversity, and the crossover and mutation probabilities are set to 0.80 and 0.20, respectively. We adopt the *keep the best* mechanism to prevent the evolutionary process from degrading. Furthermore, the tree depth of an evolved program is between 2 and 10 levels in order to avoid code bloating. To end with, the evolving process stops when the ideal individual is found, fitness value is equal to 1 or very close to the ideal (*e.g.* 10^{-6}), or the maximum number of generations is reached (*e.g.* 30).

Parameter	**Value**	**Parameter**	**Value**
Crossover rate	*0.80*	Generations	*30*
Mutation rate	*0.20*	Population size	*200*
Elitism	*keep the best*	Initial population	*Ramped half-and-half*
Tree min depth	*2*	Selection type	*Tournament*
Tree max depth	*10*	Tournament size	*7*

Table 3.2: Parameter setting of the genetic process.

The GP elected classifier, as well as the selected features and the feature fusion scheme relate only to the pair of expressions in question. The multi-class prediction using all the learned binary classifiers will be detailed in the next paragraph.

3.2.6 Multi-class facial emotion recognition

In a multi-class FER, the goal is to assign an emotion to a facial image from k possible emotions. Given the number of good performing binary algorithms in the literature and the variety of classification problems for multiple classes, there is common approaches to creating meta-algorithms using binary classifiers in order to make multi-class prediction. For simplicity reason and to reduce computational cost, the algorithm used for our experiments is the all pairs filter tree [14]. It is a tree-based algorithm that performs unique tournament elimination between sets of classes. The underlying graphical structure is a binary tree built recursively from the root as shown in figure 3.10. In k-expression FER problems, $k(k-1)/2$ binary classifiers are needed to perform a multi-class recognition. In the next section, the experimental results using the proposed $GP-FER$ method are presented.

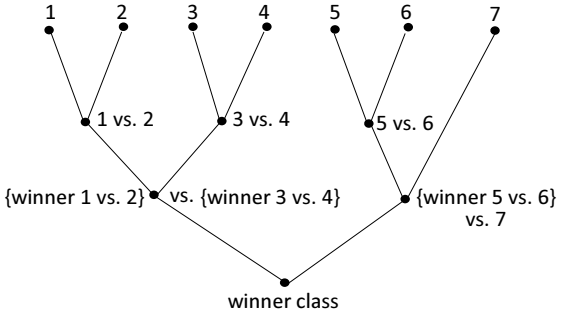


Figure 3.10: Filter tree representation.

3.2.7 Results

In this section, the proposed facial emotion recognition method is tested and compared to several relevant FER methods. The experiments have been executed on a PC with Windows 10 as an operating system with Intel®$Core^{TM}$ i7-7500U CPU @ 2.70 GHz and 8G of memory. To validate the performance of the proposed method, we conduct several experiments on four different datasets:

- Denver intensity of spontaneous facial action (DISFA) [99] contains over 1130,000 video frames of 27 adults (12 women and 15 men) ranging from 18 to 50 years old. These subjects represent different nationalities (e.g. Asian, African-American, Caucasian). This dataset represents 7 spontaneous expressions (the 6 universal expressions plus the neutral state). It scores 5 levels of intensity of 12 facial action units.

- The extended Denver intensity of spontaneous facial action database (DISFA+) is an extension of DISFA [98]. It contains a large set of posed and non-posed facial expressions data for a same group of individuals.

- Extended Cohn-Kanade CK+ dataset [**Ck+**] is considered one of the well-used datasets in the emotion recognition research community. It is a mixed (posed and spontaneous) dataset. It includes seven emotional states (the 6 basic emotions plus contempt). Indeed, CK+ provides 593 sequences from 210 adults ranging from 18 to 50 years old. Only 327 of the 593 sequences are labelled with an emotional class. Each sequence begins with the neutral expression and ends with the apex expression.

- Multimedia understanding group dataset (MUG) [2] contains sequences that begin and end with a neutral state following the onset-apex-offset temporal model. Each image sequence contains 50 to 160 images. The database includes 86 subjects with Caucasian origin aged between 20 and 35 years. There are 35 females and 51 males with or without beard. In our experiments, 325 sequences were selected.

In all experiments, we have used 10-fold cross-validation to evaluate the performance of the proposed method in order to optimize the use of available data and produce average classification accuracy results.

The proposed GP-FER method is compared to several relevant state-of-the art methods using the 10-fold cross validation protocol on the $DISFA+$, $CK+$ and MUG datasets.

Method	Dataset	Features	Classification	Expressions	Accuracy
[162]	CK+	LBP image	WMCNN-LSTM	6	97.5
[134]	CK+	Geometric+Texture	SVM	6	91.9
[134]	MUG	Geometric+Texture	SVM	6	82.9
[154]	CK+	Salient region attention	DAM-CNN	6	95.8
[44]	CK+	Salient geometric feature	SVM	6	97.8
[44]	MUG	Salient geometric feature	SVM	6	95.5
[122]	MUG	Local fisher discriminant analysis	1-Nearest-Neighbor	7	95.2
[129]	CK+	Geometric features (8 keypoints)	SVM	7	83.0
[160]	CK+	Most discriminated facial keypoints	Graph Matching	6	97.1
[77]	CK+	Facial keypoint displacement	SVM	6	99.7
[157]	DISFA+	spatial and temporal patterns	IT-RBM	7	93.0
[124]	DISFA+	LBP, LPQ, WLD, DCT	DBN-SMO	7	95.7
(GP-FER Proposed)	DISFA+	Geometric features+LBP	Genetic Program	7	94.2
(GP-FER Proposed)	CK+	Geometric features+LBP	Genetic Program	7	98.0
(GP-FER Proposed)	MUG	Geometric features+LBP	Genetic Program	7	97.2

Table 3.3: Comparison with relevant FER methods using 10-fold cross validation on ***DISFA+***, ***CK+*** and ***MUG*** datasets.

Table 3.3 presents the accuracies of facial expression recognition. The method proposed in [77] outperformed the proposed method when tested on the ***CK+*** dataset scoring an accuracy of 99.7% against 98% for the suggested method. This method requires manual localization of the facial keypoints on 6 facial expressions however the suggested method is fully automated and was tested on 7 emotions. The suggested GP-FER method widely outperformed the method presented in [134] which performs FER combining geometric and textural features. Although the authors have shown that accuracy has been considerably improved while concatenating the two types of features, the performance of this method remains relatively low compared to the proposed GP-FER. Indeed, the accuracy of the proposed method exceeds the method proposed in [134] by 6.1% and 14.3% when tested on ***CK+*** and ***MUG*** datasets, respectively. This can be explained by the fact that the proposed method performs feature selection and fusing differently for each pair of expressions however the other method concatenates features and use the resulting vector for the recognition of all the expressions. In the other hand, one of the best accuracies (97.16%), was scored by the method described in [160]. Its recognition rate has been achieved by extracting the most discriminative facial keypoints for each facial expression. This demonstrates that defining an emotion-dependent feature subset can lead to better performance as it is the case for the proposed method. For the DISFA+ dataset, the proposed $GP-FER$ method recorded an accuracy of 94.2% and outperformed the method presented in [157]. However, it was slightly outperformed by the method described in [124] using four features. Finally, the proposed $GP-FER$ method scored better or comparable results compared to many deep learning based methods such as those reported in Table 3.3 using convolutional neural networks (WMCNN-LSTM and DAM-CNN). Indeed, the suggested algorithm performs feature selection and fusion differently using binary programs evolved genetically. This ensures that the most discriminative features are adaptively selected for each pair of expressions. Actually, not all the features are significant to discriminate between all the facial expression classes. For example, the movements of the eyebrows can very well differentiate the *happiness* from the *surprise* but can mislead a classifier to select the wright expression between *surprise* and *fear*. This makes the proposed method more accurate to classify facial expressions than most of the approaches that perform feature selection globally. Moreover, unlike deep learning based methods, the proposed algorithm uses a small number of training instances to learn the binary programs and different instances are used to evolve each binary classifier. This helps prevent overfitting and the results of the cross-dataset validation experiments show that the suggested algorithm performs well (up to 91.8% accuracy) when it comes to datasets other than those on which it was trained. Be-

sides, training the suggested algorithm does not need large datasets or data augmentation, as it is the case for deep learning techniques. Indeed, to learn how to discriminate between two classes of facial emotions, the human brain does not need to be overwhelmed with instances from both classes. It only takes few instances for the brain to learn what features to select and how to fuse information from these features to perform accurate classification in unseen faces. However, to aggregate the decisions of all the binary classifiers, the all pairs filter tree algorithm was used. This algorithm uses the binary classifiers to perform a unique tournament elimination between sets of classes. The order of the binary classifiers in the tournaments can affect the classification accuracy, which constitutes a weakness of the proposed method.

3.2.8 From 2D to 3D/4D FER

Most of FER techniques, based on 2D images, performed poorly in real-world scenarios, and this is mainly due to the ignorance of the temporal information. Thereafter, many studies [164] investigated the fusion of spatio–temporal features for accurate facial expression recognition from videos. Nevertheless, even with 2D videos, many challenges (*e.g.* changing lighting conditions, head pose changing, occlusions, scale variation...) affect dramatically the performance of FER. To overcome the shortcomings of visible 2D videos, notably those related to the imaging conditions, many attempts [148] [76] tried to recognize expressions from infrared thermal images, by recording the temperature distribution formed by face vein branches. In fact, since the used sensors rely on the heat radiation emitted by the objects themselves, and therefore natural and/or artificial light sources are not required, infrared thermal images are invariant against illumination changes. Most of the proposed thermal facial expression recognition solutions are based on the analysis of the relationship between facial temperature and emotion through statistical analysis [148]. However, equipment's price, algorithm's performance and the fact that the color information is completely lost in the thermal spectrum; are the major factors that have been limiting the widespread use of thermal infrared imaging for real-world commercial applications [76]. In order to deal with the aforementioned challenges, modern 3D acquisition technologies, like laser sensors, offer a high-resolution 3D information. In fact, the problems of pose variations and illumination changes can be solved in 3D modality, which has gained growing attention over the last years. Actually, several 3D databases (*e.g.* BU-3DFE dataset [158]) were collected for facial expression and action unit recognition. However, it is still a challenging task for such systems to achieve high FER rates due to the difficulty in precisely extracting the suitable emotional features from input images [86]. These features, which are represented either statically or dynamically, can be derived from point-based geometric patterns as well as region-based appearance patterns. More recently, the advent of 4D imaging systems makes it possible to deliver 3D scan sequences of high quality for more comprehensive facial expression analysis. In addition to the shape attributes in each frame (*i.e.* static 3D scan), 4D data (*e.g.* BU-4DFE dataset [137]) also captures the quasi-periodical dynamic variations of facial expressions from adjacent frames. Thus, the fact that the human face itself is a 3D dynamic surface by nature motivates more and more the technological feasibility of studying facial expressions in 3D and 4D (dynamic 3D) spaces. In 3D and 4D FER, the most important issue is to represent shape patterns of different expressions, while expecting that the features possess high discrimination to describe the person independent geometry attributes. To deal with this issue, there are two types of expression classification. Within frame-based classification, only the current frame is used with or without a reference image (neutral face image) to recognize the expressions. Nonetheless, this classification is unable to successfully model the variability in morphological and contextual factors. Within sequence-based classification, the temporal information is employed by estimating the geometrical displacement of facial feature points between frames [43]. As a dynamic

event, recognizing facial expression from consecutive frames is more natural and proved to be more effective in recent works [93] [159]. Based on the facts above, our work within the framework of 3D/4D FER tried to recognize facial expressions by fusing a set of geometrical and appearance features of different facial regions based on 3D/4D faces. We particularly address two fundamental challenges: shape representation and feature fusion.

3.3 3D/4D Facial Expression Recognition

3.3.1 Motivation, contributions and overview

Our approach follows a four-steps process. Firstly, we compute a normalized mesh-LBP descriptor. Secondly, to fuse multiple features into a compact form independently of the number of data points, we define an LBP-based 3D covariance matrice that we called $Cov-3D-LBP$. Thirdly, we represent data as sparse conic combinations of $Cov-3D-LBP$ atoms from a learned dictionary via a Riemannian geometric approach. In fact, we adopted a Riemannian optimization method for dictionary learning and sparse coding, in which the representation loss is characterized via an affine invariant Riemannian metric. Lastly, a classifier is adopted in order to recognize the input face expression. The contributions of this paper are two-fold. To the best of our knowledge, we are the first to efficiently employ a compact combination of geometric and appearance features that is based on covariance of mesh-LBP features. Two statistical metrics, specifically LBP-Difference (LBPD) and LBP mean, are defined to compute the covariance matrix. Furthermore, the proposed descriptor is based on covariance matrices which rely on the manifold of Symmetric Positive Definite (SPD) tensors, a special type of Riemannian manifolds. The non-linear structure of these manifolds makes it impossible to use many conventional classification algorithms. Thus, we propose to learn a third-order tensor (dictionary) of basis atoms to approximate each SPD, within the $Cov-3D-LBP$ matrix, as a sparse conic combination of atoms in the learned dictionary, so that $Cov-3D-LBP$ can be approximated by a feature vector in the Euclidean space. We resort to the affine invariant Riemannian metric, instead of the Euclidean distance, in order to minimize the loss function while having feature vector representation in the Euclidean space as faithfully as possible to the Riemannian space.

The proposed method for 3D/4D FER consists of four main modules (Figure 3.11): mesh-LBP calculation and normalization, $Cov-3D-LBP$ matrices extraction using mesh-LBP features, Riemannian dictionary learning and sparse coding of $Cov-3D-LBP$ matrices, and facial expression classification. In fact, given the learned sparse code vectors, SVM classifier is used in the classification stage for static 3D faces, while HMM is exploited for 4D dynamic faces in order to predict the emotion states at different times.

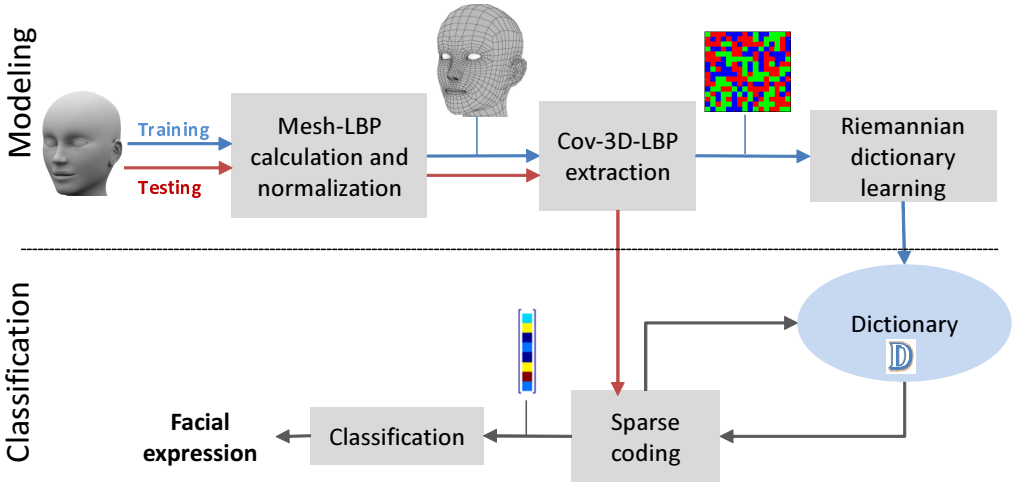


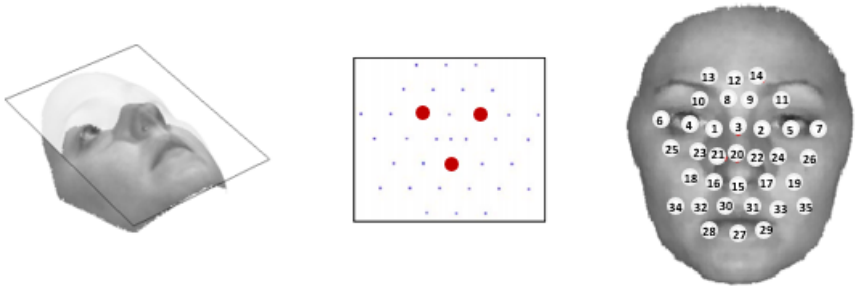
Figure 3.11: Flowchart of the proposed method for 3D/4D facial expression recognition.

3.3.2 Mesh-LBP calculation

LBP construction on triangular mesh manifolds is a recent concept [151]. In its simplest form, an LBP is an 8-bit binary code obtained by comparing a pixel's value (*e.g.* gray level, depth...) with each pixel's value in its 3×3 neighborhood. The outcome of the comparison is 1 if the difference between the central pixel and its neighbor is less than a threshold, and 0 otherwise. Obtained local description can be refined and extended at different scales by adopting circular neighborhoods at different radii and using pixel sub-sampling. In [151], LBP concept was extended to 2D-mesh manifolds by constructing sequences of facets ordered in a circular way around a central facet. The obtained structure of ordered and concentric rings around a central facet forms an adequate support for computing LBP operators (mesh-LBP). To preserve the simplicity of the original LBP, we calculated the mesh-LBP at different radial and azimuthal resolutions. In fact, we propose in this work to use the LBP binary labels instead of the decimal ones to extract the proposed covariance based features. Indeed, an LBP decimal label corresponds to its own bin in a histogram, and arithmetic operations between two or more of these labels do not necessary render meaningful semantic results. In contrast, the string of LBP binary values, obtained by comparing a central facet with its neighbors, represents each surrounding facet as a feature component, which could be more appropriate for calculating the covariance matrix. Let Ψ be a scalar function defined on the mesh, incarnating either a geometric (*e.g.* curvature) or photometric (*e.g.* color or gray level) information, the mesh-LBP operator at the facet x_c is defined as follows:

$$f_{meshLBP_{p,r}}(x_c, \Psi) = \sum_{k=0}^{p-1} \mathbb{1}_{\mathbb{R}^+}(\Psi(x_k^r) - \Psi(x_c)).\alpha(k), \tag{3.6}$$

where, r is the ring number, p is the number of facets uniformly spaced on the ring and $\mathbb{1}_\zeta$ denotes the indicator function of a subset ζ. The parameters r and p control the radial resolution and the azimuthal quantization, respectively, and the discrete function α is used for deriving different LBP variants (*e.g.* for $\alpha(k) = 2^k$, we obtain the mesh counterpart of the basic LBP operator). For the discrete surface function Ψ, we consider the Mean Curvature (MC), the Curvedness (C), the Gaussian Curvature (GC), the Shape Index (SI) as shape descriptors and the Gray Level (GL) as photometric characteristic of the facets. In the standard LBP-based face representation [1], a 2D face image is divided into a grid of rectangular blocks. Then, histograms of LBP descriptors are extracted from each block and concatenated afterwards to form a global description of the face. Given the fact that



(a) (b) (c)

Figure 3.12: Construction of the face grid on the mesh: (a) the plane, formed by the tip of the nose and the two inner corners of the eyes, is defined, (b) an ordered and regularly spaced set of 35 points are calculated on the plane from the 3 landmarks, (c) the set of points is projected on the face surface, along the plane normal direction.

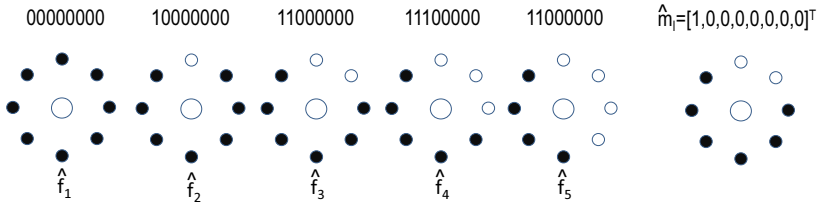
partitioning the 2D mesh manifold is not straightforward, we extract a grid of fiducial points of the face with a predefined position. Thereafter, we use their neighboring regions as local supports for computing mesh-LBP (Figure 3.12), while adopting the concatenation of separate histograms of single feature. The major problem is that the intrinsic connections, such as the correlation between features, are neglected, resulting in much information loss. Recently, covariance matrices ($CovM$) have played an important role as data descriptors in several computer vision applications. Compared with popular vector space descriptors, such as bag-of-words and Fisher vectors, the second-order structure offered by covariance matrices is particularly appealing. For instance, covariances conveniently fuse multiple features into a compact form independently of the number of data points. By choosing appropriate features, this fusion can be made invariant to affine distortions [94], and robust to static noise and illumination variations. In our case, comparatively to histogram-based descriptors, the $CovM$-based descriptors have the advantage of being more compact when taking into account the same elementary features. However, the elementary features for $CovM$ are assumed to be numerical whereas the LBPs are not, and this could lead to unstable statistics. Indeed, the LBP(-like) feature is an index of patterns and not on vector spaces. Thus, it is not theoretically reasonable to utilize the LBP(-like) feature to calculate $CovM$ in a forthright manner. For that reason, we propose an extension of the mesh-LBP, proposed in [151], by using LBPD [56] to reflect, numerically, the variations of LBPs. The proposed feature, namely $mesh - LBPD$, can be used for any central moment and is therefore inherently appropriate to be the elementary feature for $CovM$.

3.3.3 $Cov - 3D - LBP$ matrices extraction

Considering the extracted multiple elementary features, as random variables, at each pixel inside a region of interest I within an image, the $CovM$ descriptor characterizes their second order statistics inside I. In fact, covariance is a statistical measure describing the extent to which two random variables covary. Multiple elementary features extracted from each pixel x within a region I are usually arranged in a d-dimensional feature vector $f(x)$. Afterwards, a covariance matrix (2) summarizes the d^2 covariances between any couple of the d features.

$$CovM(I) = c \sum_{x \epsilon I} (f(x) - \hat{m}_I)(f(x) - \hat{m}_I)^T, \tag{3.7}$$

where, $\hat{m}$ is the mean of $f(x)_{x \epsilon I}$ and c is a normalization factor. The quadratic form ensures that $CovM$ is symmetric and positive semi-definite. In practice, $CovM$ is usually normalized (3) to a



(a) (b)

Figure 3.13: Illustration of the LBP mean. Five LBPs (a) and their mean $\hat{m_I}$ (b).

correlation matrix ($CorM$).

$$CorM(I) = \Lambda^{-\frac{1}{2}} CovM(I)\, \Lambda^{-\frac{1}{2}}, \tag{3.8}$$

where, Λ is a diagonal matrix including all diagonal entries of $CovM$. Obviously, $CorM$ is a particular form of $CovM$, whose diagonal elements are constantly equal to 1. In practice, the calculation of $CorM$ using (3) is performed by standardizing each of the d elementary features within I, in order to be zero-mean and unified standard deviation, before calculating the covariance matrix using (2). It is reported that $CorM$ achieves enhanced robustness over $CovM$, since the former disregards the standard deviations of features. Moreover, the intrinsic dimension of the covariance matrix is $d(d+1)/2$, while the one of the correlation matrix is further reduced to $d(d-1)/2$. As a result, unless otherwise specified, $CovM$ hereinafter also refers to the correlation matrix.

3.3.3.1 LBP mean

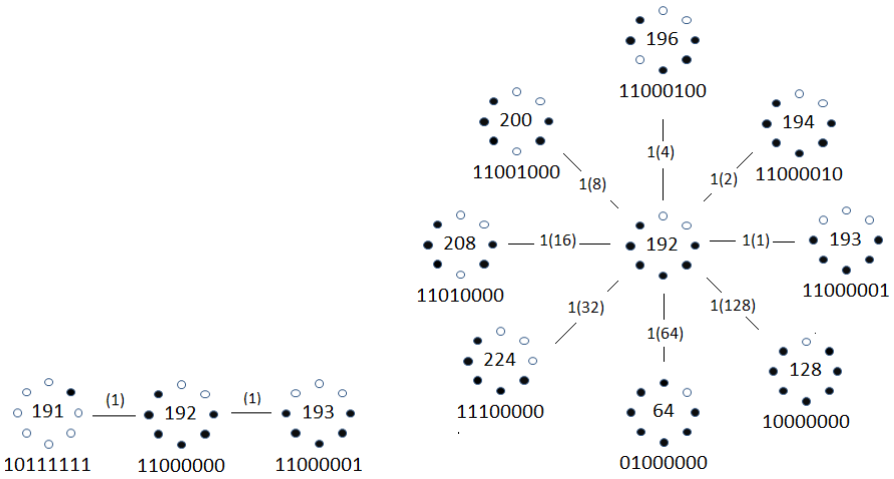
It is obvious that the difference between the feature vectors and their mean (abbreviated as difference vectors hereinafter) plays a central role in calculating $CovM$. It motivates to design a new class of LBP-like features for $CovM$ in accordance with the difference vectors. In this work, we employ the Karcher mean [70] to define the LBP mean. The Karcher mean of a set of points is the point that minimizes the summation of distances (Hamming distances in our case) to all given points. More precisely, given a set of N LBPs denoted by $S = \{f_1, \ldots, f_N\}$, the p^{th} element ($p \in \{0, \ldots, P-1\}$) of its Karcher mean $\hat{m_I}$ is defined, via a floor function $\lfloor . \rfloor$, as follows:

$$\hat{m}_I(p) = \left\lfloor \frac{\sum_{n=1}^{N} \hat{f}_n(p)}{N} + 0.5 \right\rfloor. \tag{3.9}$$

Figure 3.13 illustrates the mean $\hat{m_I}$ of five LBPs.For the integer mean in (4), $\hat{m_I(p)} = 1$ means that the majority of LBPs under consideration have a value of 1 in the p^{th} bit, and vice versa.

3.3.3.2 LBP difference

The Euclidean distance, which is implied by the arithmetic subtraction of two numerical values, fails to point out the precise relation of patterns. In the general case, a unit distance between two points usually indicates that they are semantically similar. However, it is not always the case for LBPs when it comes to Euclidean distance (Figure 3.14a). Moreover, the responses of LBPs rely on the subjectively assigned order of bits, *i.e.* the way in which the bits are assigned with the power weights from the most one $2^{(P-1)}$ to the least one 2^0 (= 1). Statistics, such as the covariance between the LBP and other features, are inherently dependent on the order of bits. As a result, there is a potential risk for the covariance between the LBP of different features to be affected by noise and abrupt changes,



(a) (b)

Figure 3.14: Euclidean distance *vs.* Hamming distance for comparing two LBPs (*e.g.* neighbors having a unit distance to the LBP '192'): (a) neighbors under Euclidean distance, (b) neighbors under Hamming distance. Numbers inside (*resp.* outside) parentheses denote the Euclidean (*resp.* Hamming) distances between the two connecting LBPs.

especially for the most significant bits (with large weights). In order to overcome the aforementioned problems, we adopted the Hamming distance to evaluate the distance between two LBPs (Figure 3.14b). Instead of considering the input as numerical numbers, the Hamming distance regards the input as binary strings and then aggregates the bitwise differences. It is formally defined (5) as the count of bits that are different between two binary strings a and b of the same length P.

$$d_H(a,b) = \sum_{p=0}^{P-1} (a(p) \oplus b(p)), \tag{3.10}$$

where, $a(p)$ and $b(p)$ ($p \in \{0,\ldots,P-1\}$) are the p^{th} bit of a and b, respectively, and $\oplus(.)$ denotes the exclusive OR operator. It is clear that the Hamming distance reflects the topology of the space of LBPs more precisely than the Euclidean distance. In addition, since the Hamming distance is calculated bit by bit regardless of the bits' weights, only a fraction of Hamming counts are affected by abrupt changes in images in most of the cases. Hence, it is believed that measuring the proximity of LBPs by the Hamming distance is more robust against the noise. The Hamming distance motivates to avoid the weighting in (1) and only preserves the co-occurrence of local comparisons. Thus, the local pattern is considered as a binary vector $\hat{f}_{meshLBP}$ in this work, and each element of this vector corresponds to a particular bit of the ordinary meshLBP. Hence, the p^{th} bit of $\hat{f}_{meshLBP}$ is defined by:

$$\hat{f}_{meshLBP}(p) = \mathbb{1}_{\mathbb{R}^+}(\Psi(X_p) - \Psi(X_c)), \tag{3.11}$$

where, X_p ($p \in \{0,\ldots,P-1\}$), are P neighboring pixels surrounding the central pixel X_c. So, the Hamming distance between $\hat{f}_a$ and $\hat{f}_b$ can be expressed by:

$$d_H(\hat{f}_a - \hat{f}_b) = \sum_{p=0}^{P-1} \hat{f}_a(p) \oplus \hat{f}_b(p). \tag{3.12}$$

Given the LBP mean $\hat{m}_I$, we can easily obtain the LBP difference vector through $\hat{d} = \hat{f} - \hat{m}_I$. The magnitude of the expected features is supposed to reflect how $\hat{f}$ and $\hat{m}$ are different. To this

end, motivated in part by Hamming distance in (7), we reach the norm of $\hat{d}$ to aggregate the bitwise differences between $\hat{f}$ and $\hat{m}$. Specifically, let $\hat{f}$ denotes $\hat{f}_{meshLBP}(x, \Psi)$ in (1) for clarity, the magnitude of the expected features is defined by:

$$f^{u}_{meshLBPD}(x, \Psi) = \left\| \hat{f} - \hat{m_I} \right\|, \tag{3.13}$$

where, $\|.\|$ can be any type of norms defined in vector spaces, such as the L^1 norm and the L^2 norm. We call the feature defined above as the mesh-LBP Difference (mesh-LBPD), and its response is non-negative in accordance with the positivity property of norms. In summary, as a variant of mesh-LBP, the mesh-LBPD is numerical, so it can be directly applied to calculate the $CovM$. Thus, once $LBP\ mean$ and $LBPD$ are estimated, the diagonal entries of $CovM$, that we called $Cov-3D-LBP$, represent the variance of each feature and the non-diagonal entries represent their co-variations.

3.3.4 Riemanian dictionary learning and sparse coding

The extracted $Cov{-}3D{-}LBPs$ are encoded as symmetric positive definite (SPD) matrices. While these matrices form an open subset of the Euclidean space of symmetric matrices, viewing them through the lens of non-Euclidean Riemannian geometry often turns out to be better suited in capturing several desirable data properties. In particular, Dictionary Learning and Sparse Coding (DLSC) of SPD matrices has received significant attention in the vision community due to the performance gains it brings to the respective applications [96]. In fact, sparse learning represents the target variable as a sparsely linear combination of a set of basis functions. Thus, to associate facial expressions with the extracted visual features, a weighted multi-modal shared sparse learning can be used to automatically learn the combination coefficients [165], in order to predict the probability distribution of an unseen image by linearly integrating the ones learned from the training data. Given a training set Y, DLSC defines a dictionary B of basis atoms, such that each data point $y \in Y$ can be approximated by a sparse linear combination of these atoms, while minimizing a suitable loss function. Formally, the DLSC consists in resolving the optimization problem below:

$$\min_{B, \theta_y \forall y \in Y} \sum_{\forall y \in Y} L(y, B, \theta_y) + \lambda Sp(\theta_y), \tag{3.14}$$

where, the loss function L measures the approximation quality obtained by using the "code" θ_y, while λ regulates the impact of the sparsity penalty Sp. Let $Y = \{ Y_1, \ldots, Y_N \}$ denotes a set of N SPD matrices, and M^d_n is the product manifold obtained from the Cartesian product of N SPD manifolds, our goals are to learn a third-order tensor (dictionary) $B \in M^d_n$ where each slice represents a dictionary atom B_j; and to approximate each Y_i as a sparse conic combination of atoms in B (*i.e.*, $Y_i \sim B\alpha_i$ where $\alpha_i \in \mathbb{R}^n_+$ and $Bv := \sum_{i=1}^{n} v_i B_i$ for an n-dimensional vector v). Thus, our joint DLSC objective is given by:

$$\min_{\alpha \in R^{n \times N}_+, B \in M^d_n} \frac{1}{2} \sum_{j=1}^{N} d^2_R(Y_j,\ B\alpha_j) + Sp(\alpha_j) + \Omega(B), \tag{3.15}$$

where, Sp and Ω are regularizers on the coefficient vectors α_j and the dictionary tensor. Since Euclidean distance can not evaluate precisely the proximity of $CovMs$ [145], we adopt the Affine Invariant Riemannian Metric d_R (11).

$$d_R(X, Y) = \sqrt{\sum_{i=1}^{d} ln^2(\lambda_i(M_1, M_2))}, \tag{3.16}$$

where, $\{\lambda_i(X,Y),\ 1 \leq i \leq d\}$ are the d generalized eigen-values of two positive definite matrices X and Y. Note that the Riemannian metric provides a measure for computing distances on the manifold, and given two points on the manifold, there are infinitely many paths connecting them, of which the shortest path is termed the geodesic (12), using Frobenius norm.

$$d_R(X,Y) = \left\| \log(X^{-1/2} Y X^{-1/2} / \right\|_F. \tag{3.17}$$

Assuming that the coefficient vectors α are available for all matrices, the updating of the dictionary atoms can be separated from (10) and written as follows:

$$\min_{B \in M_n^d} \Theta(B) := \frac{1}{2} \sum_{j=1}^{N} \left\| \log(Y_j^{-\frac{1}{2}} (B\alpha_j) Y_j^{-\frac{1}{2}} \right\|_F^2 + \Omega(B). \tag{3.18}$$

Referring back to (10), and considering the sparse coding subproblem, our objective returns to solve, for a data matrix X_j, the following optimization problem, while supposing the availability of a dictionary B:

$$\min_{\alpha_j \geq 0} \phi(\alpha_j) := \frac{1}{2} \left\| \log \sum_{i=1}^{n} \alpha_j^i X^{-\frac{1}{2}} B_j Y^{-\frac{1}{2}} \right\|_F^2 + Sp(\alpha_j), \tag{3.19}$$

where, α_{ij} is the i^{th} dimension of α_j and Sp is a sparsity inducing function. For simplicity, we use the sparsity penalty $Sp(\alpha) = \beta||\alpha||_1$, where $\beta > 0$ is a regularization parameter. Since α is positive in our case, we replace this penalty by $\beta \sum_i \alpha_i$. Assume that we have a dictionary B composed of $Cov - 3D - LBP$ matrices $\{B_1, B_2, \ldots, B_N\}$ as atoms and an input matrix Y that needs to be sparse coded, the goal of Riemannian sparse coding for matrix Y is to seek a non negative sparse vector $a = [\alpha_1, \alpha_2, \ldots, \alpha_N]^T$, which makes the linear combination $\sum_i \alpha_i B_i$ as close to Y (in Riemannian geodesic distance) as possible. The above sparse coding problem can be defined as follows:

$$\min_{\alpha \geq 0} \phi(\alpha) := \frac{1}{2} d^2 \left(\sum_{i=1}^{N} \alpha_i B_i, Y \right) + Sp(\alpha), \tag{3.20}$$

It has been proved in [19] that $\varphi(\alpha) = d^2(\sum_i B_i, Y)$ is a convex function on the set A (16).

$$A := \left\{ \alpha \mid \sum_{i=1}^{N} \alpha_i B_i \leq Y, \, and \, \alpha_i \geq 0 \right\}. \tag{3.21}$$

Using the $\| \cdot \|_1$ norm as the sparsity penalty, we can rewrite (15) as the following minimization function via replacing the distance by the d_R:

$$\min_{\alpha \geq 0} \phi(\alpha) := \frac{1}{2} \left\| \log \left(\sum_{i=1}^{N} \alpha_i Y^{-\frac{1}{2}} B_i Y^{-\frac{1}{2}} \right) \right\|_F^2 + \gamma \left\| \alpha \right\|_1, \tag{3.22}$$

where, γ is a regularization parameter. The above minimization problem with the constraint condition (16) is a regularized non-negative convex optimization problem that we solved using the spectral projected gradient [34].

3.3.5 Classification methods used for 3D FER and 4D FER

For 3D classification, generated features from the sparse coding are used to feed six SVMs, each one with a radial-basis kernel. In fact, the SVM classifier has been shown to be effective, for facial expression classification, particularly for small amount of training data [11]. One-vs-all scheme is used to train one SVM for each facial expression. In this case, positive examples come from one facial expression, while negative examples come from all the other expressions. On the other hand, in the case of 4D FER, the inputs are sequences of 3D frames that constitute the temporal dynamics to be classified. In our case, each expression exp ($expr \in \{anger, disgust, fear, happiness, sadness, surprise\}$) is modeled by an HMM M_{expr}. We adopted the HMM model proposed in [123] which has proved its effectiveness in clustering the expressive states of a sequence. In this model, four states are used to represent the temporal behavior of each expression (Figure 3.15). It is stipulated that each expression sequence starts and ends with a *neutral* expression (S1). The expression reaches its highest intensity at the frame captured by the *apex* state (S3). Intermediate frames, where the expression of the face changes from neutral to very expressive and inversely, are captured by the *onset* (S2) and *offset* (S4) states, respectively. Figure 3.16 shows a 4D sequence capturing a sequence of frames illustrating the temporal dynamic aspect of the happiness expression through a sample subject face. To train each HMM, expression sequences are considered as observation sequences $Y = \{y_1, y_2, ..., y_T\}$, where each observation y_t at time t is given by the feature vector $A^t = \{a_1^t, ..., a_K^t\}$. Then, a vector quantizer is designed to cluster feature vectors of the training sequences into a reproduction alphabet [89]. This provides a mapping between multidimensional feature vectors, taking values in a continuous domain, with the alphabet of symbols emitted by the HMM states. Each M_{expr} is initialized with random probabilities and the *forward-backward* algorithm [120] is applied to train the model, while finding the maximum likelihood estimate of its parameters given the training sequences. Thus, for an unseen 3D sequence, the corresponding observation Y is fed to the six HMMs and the *Viterbi* algorithm is used to determine the most likely sequence of states (*Viterbi* path) $X = \{x_1, ..., x_T\}$, which corresponds to the state sequence recording the maximum of likelihood to Y. Finally, the input sequence is classified as belonging to the $expr$-class corresponding to the M_{expr} whose log-likelihood along the best path is the greatest one.

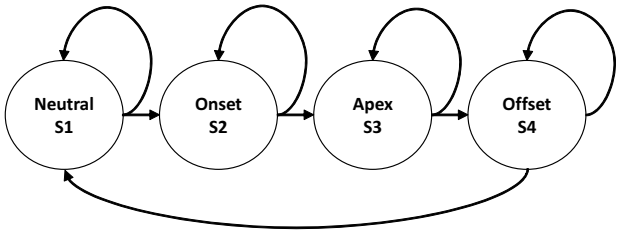


Figure 3.15: Structure of the HMM of an expressive sequence.

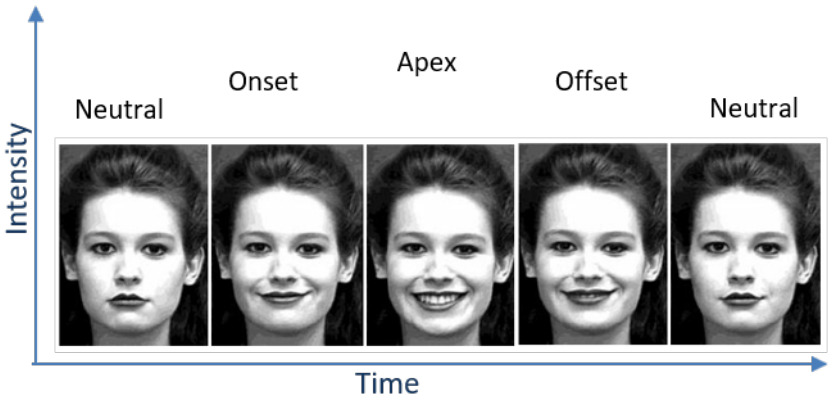


Figure 3.16: Frames extracted from a dynamic 3D video sequence illustrating the temporal dynamics of the happiness expression (the four states of the HMM are depicted in the sequence).

3.3.6 Results

To validate the effectiveness of the proposed method for 3D/4D FER, we conducted extensive experiments on the challenging BU-3DFE and BU-4DFE databases. For the validation of 3D FER, three main protocols are generally used in the literature to evaluate the FER methods on the BU-3DFE dataset. Early works chose 60 subjects and averaged the accuracies of one or two rounds of 10-fold cross-validation, totally with 10 or 20 times of train and test sessions. This protocol is denoted herein as *ProtI*. Later, authors in [46] suggested to choose 60 subjects and averaged the accuracies of 100 rounds of 10-fold cross-validation, resulting in 1000 times of train and test sessions in total. This protocol is indicated herein by *ProtII*. Within the last protocol, denoted as *ProtIII*, 60 subjects are randomly selected in each round of 10-fold cross-validation and the accuracies of 100 rounds are thereafter averaged.

Figure 3.17 presents the results of comparing the proposed method (PM) performance against relevant methods from the state-of-the-art. It is clear that the proposed method records the highest recognition performance (96.8%), using *ProtI*, followed by [166] (96.4%) and [142] (95.1%).

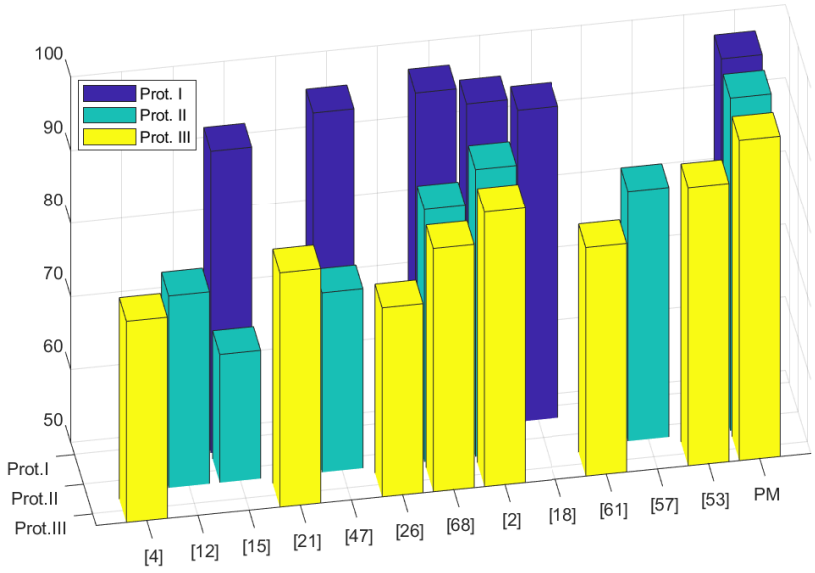


Figure 3.17: Comparison of the proposed method (PM) *vs.* state-of-the-art methods ([13], [46], [49], [60], [142] [83], [166], [7], [51], [161], [156] and [150]) for posed expressions in BU-3DFE dataset.

Moreover, we conducted experiments to evaluate the effectiveness of the mesh-LBPD feature and the $Cov-3D-LBP$ descriptor within the task of 3D FER. For comparison purpose and reporting results, we used the classification results obtained by the SVM classifier. The comparison was realized on the BU-3DFE dataset, while using $ProtIII$.

At the meantime, we provide a comparative study of the proposed method, against the relevant ones in literature, while taking into account different protocols on the BU-4DFE dataset (Table 3.4). In this table, $\#E$ denotes the number of expressions, $\#S$ is the number of subjects, $\#\text{-}CV$ provides the number of cross-validations, and "Full Seq/Win" means decision is made based on full sequence (Seq) or on sub-sequences captured by a sliding window Win. Overall the dataset, [10] reported an average performance of 93.8% using HMM. In their experiments, a sub-sequence of a constant window width including 6-frames (Win = 6) was used. Studies conducted by [137] and [138] achieved ones of the highest accuracies when using a sliding window of 6 frames. Nevertheless, these methods require manual annotation of 83 landmarks on the first frame. In addition, the dense tracking scheme is time consuming. For the same studies, it is worth noting that the problem of the window-based evaluation protocol resides in labeling all the sub-sequences from the neutral intervals as one of the six expressions, which influences considerably the final results. For that reason, we classified the entire sequences, similarly to [82], [155], [166] and [167]. Moreover, the method of [135] was evaluated on only the three simplest expressions (HA, SA and SU) and displayed the performance of 97.75%. Compared to the works of [82] [87] [167] [166] [7] that conduct classification under the same protocol (*i.e* fully automated, 6E, 60S, 10-CV and "Full seq"), the proposed method achieves encouraging accuracy (= 94.2%). This performance is accomplished thanks to the discrimination power of the $Cov-3D-LBP$ descriptors and to the accurate classification performed by the HMM.

3.3.7 Discussion

The $Cov-3D-LBP$ descriptor has high discriminative power through blending multiple features. Moreover, it has good robustness derived from the following aspects. Firstly, it is robust to abrupt

noise when the number of facets in a face region is large enough to form stable statistics. Secondly, the robustness can be enhanced if all selected elementary features have the consistent robustness (*i.e.* when only features insensitive to rotation are selected, the resulting $Cov-3D-LBP$ descriptor is robust to rotation changes). Thirdly, normalization usually leads to improved robustness by eliminating the affects coming from different variances of features. In addition, the $Cov-3D-LBP$ is a compact descriptor of low dimensionality thanks to the correlation matrices. In our case, since $Cov-3D-LBP$ was defined for five features, the intrinsic dimension of the descriptor in one face region is only 10 and for the whole face (35 regions) is 350. Low dimensionality generally leads to two benefits: low storage requirements and reduced computational complexity. In contrast, commonly used histogram-based descriptors always have much higher dimension. In fact, covariance-based representation leverages the integral representation, which leads to efficient computation cost [95] [155]. Besides, fast approximation of the Riemannian metric speeds further the computation of the distance between two $Cov-3D-LBP$s [153].

Method	**Classifier**	**Experimental settings**	**Accuracy**
[137]	HMM	6E, 60S, 10-CV, Win=6	90,4
[10]	Random Forest	6E, 60S, 10-CV, Win=6	93,8
[138]	HMM	6E, 60S, 10-CV, Win=6	**94,4**
[155]	NN	6E, 60S, 10-CV, Full seq	78,8
[82]	HMM	3E, 60S, 10-CV, Full seq	92,2
[167]	SVM	6E,60S, 10-CV, Full seq.	93,39
[167]	HMM	6E, 60S, 10-CV, Full seq.	94.18
[135]	CRFs	3E, 60S, 10-CV, Full seq.	**97.75**
[166]	HMM	6E, 60S, 10-CV, Full seq	87,1
[7]	HMM	6E, 60S, 10-CV, Full seq	90,4
[87]	DGIN	6E, 60S, 10-CV, Full seq	92,22
PM	HMM	6E, 60S, 10-CV, Full seq.	94,2

Table 3.4: Comparison of the proposed method (PM) *vs.* state-of-the-art methods for posed expressions in BU-4DFE dataset.

Generally, FER results on BU-3DFE and BU-4DFE databases show that the expressions of HA and SU are well identified, while AN, SA, and DI have moderately lower recognition rates. However, FE records the lowest average recognition rate, which is the case for all works on 3D/4D FER. A probable reason is mainly that FE is not defined as well as HA and SU for human beings, and for some individuals, certain expressions are difficult to perform. This confirms that it is not sufficient to recognize FE using only static information. Moreover, when dynamic cues are added, the classification rate of FE is ameliorated, highlighting the necessity of 4D data.

3.4 Conclusion

In the framework of 2D FER, a robust method for automatic facial expression recognition, is proposed. Genetic programming-based binary programs, which incorporate feature selection and fusion in the learning process, are proposed to discriminate between pairs of expression classes. The overall expression recognition is performed using a unique tournament elimination between the learned binary classifiers. The suggested method selects and combines differently linear, eccentricity and LBP

features for each pair of expressions. This allows to choose the most discriminating subset of features separating each pair of classes and to fuse them while avoiding information redundancy, thanks to GP optimization. Unlike deep learning-based methods, which need data augmentation to perform training phase, the suggested method works well with limited training instances. The reported performances were among the best results recently found for posed and spontaneous facial expression recognition. Improvements can still be made to the GP-FER method. Indeed, the proposed method was only tested on the combination of three geometric and appearance features. Other features can be tested, and their integration can be easily achieved by simply defining the corresponding terminal functions within the selection layer. In the framework 3D/4D FER we presented an automated method, which is based on $Cov-3D-LBP$ descriptor. The $Cov-3D-LBP$ descriptor combines the mesh-LBPD of different features within a covariance matrix. In fact, in order to compose a powerful $Cov-3D-LBP$ matrices that capture shape as well as texture characteristics, the proposed method fuses the mesh-LBPD from the mean curvature, Gaussian curvatures, the curvedness, and the shape index , in addition to the gray level value. This permits to improve the distinctiveness in deformation of facial regions of different expressions. Then, DL-SC has been established to capture several desirable data properties by extracting robust and discriminative descriptors from the $Cov-3D-LBP$. For expression label prediction, SVM and HMM are used in the case of 3D and 4D, respectively. We carried out extensive experiments on both the BU-3DFE and the BU-4DFE challenging benchmarks, while comparing the obtained results to those performed with relevant state-of-the-art methods. Recorded performances demonstrate the effectiveness of the proposed method.

Chapter 4

Breast cancer diagnosis from mammographic images

The main results presented in this chapter have been published in the following international journal: Computers in Biology and Medicine (CIBM 2021) [41]

4.1 Introduction

Breast cancer is the most common form of cancer among women. Computer-Aided Diagnosis (CAD) systems are very useful for giving radiologists a second opinion to take decisions swiftly. The decision support provided by CAD systems can be explicit classification or Content-Based Mammogram Retrieval (CBMR). Indeed, differently from automatic classification tools, CBMR provides a sorted set of similar images with a confirmed diagnosis relative to a given mammogram lesion. Retrieved images can serve for supporting radiologists in the diagnosis as well as for educational purposes, while providing more explainable results. Both classification and CBMR can follow the same pipeline for feature extraction, selection and fusion. Furthermore, the diagnosis of breast cancer from mammograms can be seen as a problem of texture classification where the CAD system should differentiate between texture of normal tissues and that of cancerous tissues. Thus, various texture classification techniques have been proposed [65, 28]. Some works tried to use descriptors which are known to be efficient in texture classification, notably the Local Binary Pattern (LBP) descriptor, in order to classify breast tissues as cancerous or normal [78, 91]. LBP descriptor uses the signs of the differences between a pixel and its immediate neighbors in order to represent the texture locally [112]. This encoding, which exploits the signs of the differences while ignoring the magnitudes, has shown very good results for classifying textures [54]. However, the problem with mammograms is that the textures of normal tissues and cancerous tissues are hardly distinguishable. As shown in Figure 4.1, the LBP transforms of three Regions Of Interest (ROI) representing benign, malignant and normal tissue as well as the corresponding LBP histograms do not represent any specific difference for one class compared to another. Therefore, using a technique that has been proven in texture classification may not be sufficient for an accurate diagnosis of breast cancer from mammograms [78]. In fact, as much local information as possible must be exploited to distinguish two fairly similar texture classes [53]. In addition, the global feature vector must be constructed in such a way as to retain information that highlights the difference between malignant tissues and normal ones [26]. Therefore, the way to aggregate the local information to construct the global tissue description should be automated. Indeed, classical concatenation of local representation, commonly used with handcrafted features, can lead to information loss [38]. The goal of this work is to propose a different way for texture analysis for more accurate breast cancer diagnosis

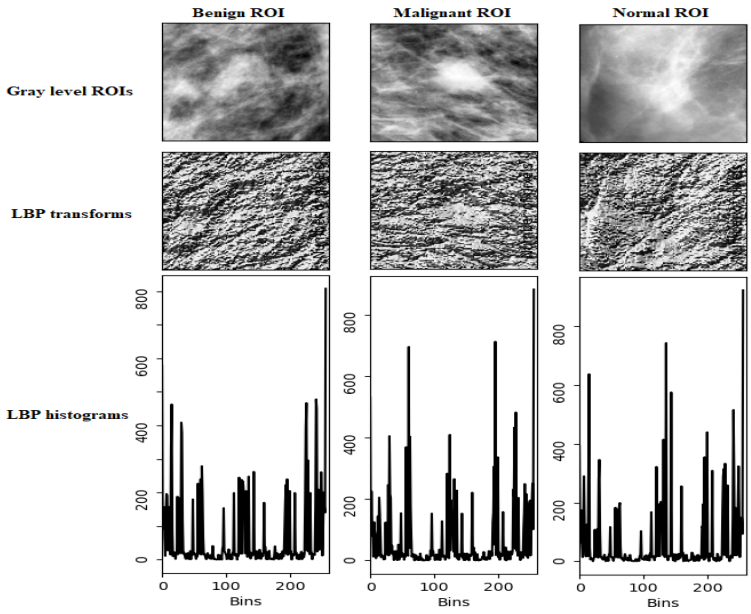


Figure 4.1: Benign, malignant and normal ROIs (of size 128×128) from the DDSM dataset and their corresponding LBP transforms and LBP histograms.

from mammogram ROIs. This can be accomplished by acting differently when representing texture locally and globally [40]. Locally, it relates to finding an efficient way to encode as much local texture information as possible without exploding dimensionality. Globally, the objective is to automate the construction of the overall texture description so that the characteristics discriminating malignant and normal tissues can be preserved. Thus, in order to enhance accuracy for CAD techniques for breast cancer diagnosis, we investigate the use of an LBP-like texture representation that uses both signs and magnitudes of the differences between the central pixel and its neighbors in order to describe tissue locally. We propose to learn automatically a texture descriptor that uses this local texture representation to generate an overall feature. For this purpose, genetic programming techniques are investigated to generate a descriptor that produces discriminative features in order to facilitate the task of a supervised classifier for taking reliable decisions. The research gap filled by the suggested method lies in the proposition of an evolutionary context-aware descriptor able to select automatically the most appropriate and restrictive set of features before fusing them depending on the specificity of the classification context without the need for a pre-processing step. Thus, the main contributions of this study are threefold:

- To the best of our knowledge, we are the first to propose an evolutionary-based descriptor for generating robust features to diagnosis breast cancer from mammogram ROI images. Moreover, a fitness function based on the Fisher Separation Criteria (FSC) has been proposed in order to learn descriptors that generate the most discriminative features considering intra-class as well as inter-class characteristics.
- We represent ROI texture locally using LBP-like difference of gray level magnitudes. But unlike the LBP operator, the proposed descriptor uses both magnitudes and signs to generate the feature vector.
- The suggested method is fully-automated but, unlike deep learning-based methods, it does not need a large training set to perform accurate classification and retrieval.

The rest of this chapter is organized as follows. In Section 4.2, we present a brief and exhaustive literature review on relevant existing methods for texture-based methods of cancer diagnosis from mammographic images and we propose a taxonomy to classify the studied works. The proposed method is detailed in Section 4.3. The results are presented in Section 4.6. Outcomes and findings of the suggested method are discussed in Section 4.7. In Section 4.8, we conclude the proposed work and present some ideas for future studies.

4.2 Literature review and proposed taxonomy

Texture is the most important visual cue for describing breast tissues from mammography ROIs, since it illustrates discriminative information about tissue property. The extracted textural features from mammography images can be categorized into handcrafted features and deep learning-based features. On the one hand, within the framework of handcrafted features widely used for breast cancer CAD, features can be categorized into classical texture, curvlet-based, nature-inspired and uncertainty-based. Within the category of classical texture-based features, Kral et al. [78] have proposed a texture-based method that uses uniform Local Binary Pattern (μLBP) to classify mammography images into normal and cancerous cases. Each image has been divided into cells before calculating a 59-sized feature vector representing the histogram of uniform patterns for each cell. The concatenated feature vector is fed to an SVM classifier in order to label the input image. The best accuracy reported among the tested datasets is 84%, which remains a relatively low rate especially when it comes to make a critical decision. In [65], authors have presented a CAD system for the classification of breast masses within mammographic ROIs into malignant and benign. Texture has been described using thirteen features based on the Gray Level Co-occurrence Matrix (GLCM). The extracted features are fed to an SVM classifier and the method has achieved an accuracy of 94%. Nevertheless, the method was evaluated only on 50 ROIs. More recently, the authors in [57] have performed first order statistical and GLCM-based textural feature extraction in order to detect breast masses from preprocessed and segmented images. The classification using a KNN classifier has achieved 92% accuracy. In [5], shape, statistical and textural descriptors from ROI images have been investigated in order to extract fourteen features that are thereafter fed to a non-linear SVM classifier. An accuracy of about 90% has been obtained in multi-class breast tissue classification (normal, benign and malignant). However, a pre-processing step is required to remove unwanted noise artifacts and pectoral muscle from mammograms. In [147], the Canny edge detector followed by the Hough transform have been applied and subsequently local texture characteristics are extracted. Four types of intensity-based features have been employed (mean, entropy, standard deviation and variance), which are then fed to the classifier for training. An accuracy of 94% has been achieved by this method for the classification of breast tissues into normal and abnormal. Differently, the authors in [109] have used six normalized first order features (mean, standard deviation, smoothness, third moment, uniformity and entropy) that are fed to a KNN classifier in order to detect the presence of breast masses. The best accuracy achieved by this method is 91.8%. In [75], a micro-calcification detection method has been introduced using a voting classification based upon SVM, KNN and decision trees. The authors have extracted LBP, Tamura, wavelet and Discrete Cosine Transform (DCT) as features. Different combination of classifiers and extracted features have been tested during the experimental phase and KNN has recorded the best accuracy in almost all cases. Three types of segmentation techniques have been investigated on the preprocessed mammograms which are: top hat transformation, watershed method under constraint of markers obtained from white hat method and split and merge method with homogeneity criteria based upon mean, variance and uniformity. In [103], Haralick's features have been extracted from the ROI images, before applying the Kernel Principal Component Analysis (KPCA) in order to reduce

the size of the obtained feature vector. Finally, a wrapper-based parameter optimized Kernel Extreme Learning Machine (KELM) is used to select the most prominent features from the reduced feature vector. A multi-class classification accuracy of 92.61% has been reported on the Digital Database for Screening Mammography (DDSM). In [81], the authors proposed to reduce false positives using texture and shape features and the bagged trees classifier after performing a heavy pre-processing step followed by abnormality segmentation. The method achieved overall accuracy of 93.15% on the DDSM dataset. In [28], detection of suspicious masses is performed using gray difference weight and Maximally Stable External Regions (MSER) detector. The reported classification accuracy on the Mammographic Image Analysis Society digital mammogram database (MIAS) has reached 94.66%. However, the detection step is preceded by background suppression, pectoral muscle segmentation and breast region segmentation.

Curvelet transform has also been used within the context of breast cancer diagnosis from mammoraphic images [101, 102]. Indeed, this transform provides an efficient representation of smooth objects with discontinuities along curves by representing image at different scale and angles. However, curvelet coefficient-based methods suffer from the curse of dimensionalty. To overcome this problem, authors in [26] have proposed a feature extraction method based on moment theory. The best accuracy for malignancy detection (=81.35% on the DDSM dataset) has been achieved using four first-order-based features with a KNN classifier. In [72], curvlet features are extracted from the curvelet coefficients, by applying gray level co-occurrance matrix, and then combined with the regional properties of the preprocessed images in order to detect breast cancer from ROI images. More recently, authors in [71] have proposed a combination of statistical, intensity, geometry features and texture features, extracted from curvelet coefficients, in order to enhance the classification accuracy. Almost all of the methods that tried to detect breast cancer from mammography images, using handcrafted texture-based features, have performed a prepocessing step and a segmentation step, which are heavy and time consuming tasks. Despite this, the accuracy of these methods remains relatively low, except for some of them which did not perform extensive experimentation to assess their effective performance.

To enhance the performance of handcrafted features, some works have used nature inspired techniques. For instance, in [79], multilevel image thresholding based on Otsu's method is considered for the detection of suspicious mass lesions. To tackle the difficulty of defining the threshold value when the intensity profile in and around the anomalies does not vary much, a nature-inspired wind driven optimization was used. Differently in [104], a fusion-based feature extraction method is used to combine 2D block discrete wavelet transform and GLCM, while employing PCA in order to reduce the size of the obtained large feature vector. Forest optimization algorithm, which is an evolutionary algorithm, is then adopted as a wrapper-based technique to perform both feature selection and classification. Artificial bee colony and whale optimization are also used in [136] for simultaneous feature subset selection and parameter optimization of an artificial neural network for breast cancer diagnosis. Other methods have used uncertainty on texture features to deal with the problem of breast cancer detection. For instance, the method in [22] has proposed the formulation of five feature types by extending the information set to encompass the concept of an intuitionist fuzzy set. The certainty of the pixel intensities of mammograms to a class and the deficiency in the fuzzy modeling referred to as the hesitancy form a pervasive information set used to extract five feature types termed as probability-based pervasive information set features. Fuzzy logic and probabilistic neural network were used in [52] for breast tumor classification into normal and abnormal, which led to raising the system efficiency by increasing the accuracy rate up to 99% and reducing the computational time as reported by the authors.

On the other hand, various deep learning-based methods have been proposed to diagnosis breast

cancer from mammography images. These methods can be categorized into models built from scratch, transfer learning-based methods and methods that use deep features with classical classifiers. In fact, first works tried to build deep models from scratch as in [18] where authors have proposed a 9-layered convolutional neural network for binary and multi-classification of breast cancer. Although morphological closing and masking are performed for noise removal and ROI segmentation, the accuracy has only reached 65%. In [118], authors have proposed a CNN-based framework, called BC-DROID, which performs ROI detection and diagnosis in a single step by training it first on physician-defined ROIs and then on full mammogram images. The accuracy of the classification is up to 93.5%, however this accuracy is decreased by the ROI detection rate which is 90%. In [125], a deep encoder-decoder CNN-based architecture including 23 layers has been introduced for cancer and abnormality detection. Histogram equalization has been applied to improve the contrast of the input images which are then randomly rotated for data augmentation. The model has achieved classification accuracy of 94.31% for cancer diagnosis and 95.01% for abnormality detection. In [85], four CNN-based models have been investigated in order to study the impact of depth and hidden layers' structures on model performance while classifying abnormalities and benign vs. malignant. The best performing model is four convolution layered with dropout of 0.7, called CNN-4d, with an accuracy of 89.05%. More recently, authors in [33] proposed a CAD system for cancer detection based on two phases. In the first phase, they performed heavy pre-processing including format unification, noise removal, image enhancement, ROI extraction, augmentation and image resizing. In the second phase, they proposed a CNN model from the scratch to learn features and classify the breast lesions in mammogram images. The results were encouraging when tested on two datasets. Overall, the performance of the proposed deep models highly relies on the quantity of the training data. Most of them need large scale datasets to ensure generalizability. Therefore, the availability of large amount of labeled breast images is mandatory for efficient training of these models. However, most of publicly available mammography datasets do not meet this constraint. Indeed, learning over small sized datasets may lead to insufficient performance. Some of the proposed deep learning-based methods have tried to overcome this problem by augmenting data in order to provide models with more training data. But, using the same modified breast images can lead to overfitting and dataset dependency. Therefore, some works have resorted to the use of regularization techniques by fitting the number of layers and adapting the size of the filters [85]. Transfer learning is also explored in [47], where a pre-trained VGG-16 model has been adopted to extract features from mammograms before using these features to train a neural network classifier. Then, the model has been fine-tuned by updating weights in several final layers using back propagation. However, fine-tuning the model increases the accuracy by only 0.8% while being 95% more costly in terms of computational time. The authors have stated that they tested the obtained model before the fine-tuning and obtained about 90.5% accuracy in abnormality detection in DDSM dataset without obvious overfitting, but this was not proven experimentally. In [36], MobileNet and NasNet architectures have been explored for their suitability to breast cancer diagnosis. The performance of these networks has been compared with two pretrained networks, namely Inception-V3 and ResNet50, and fine tuning of the pretrained networks has been performed using mammographic images taken from the CBIS-DDSM database. It is worth noting that these networks are used only to extract features by tuning the weights of the last classification layer while freezing the weights of other layers. The best accuracies reached are 74.3% and 78.4% which were achieved by MobileNet and ResNet50, respectively, on preprocessed and segmented mammographic images. In [121], various deep features are extracted using deep convolutional neural networks and fed to a support vector machine classifier. The results on the DDSM and the MIAS datasets were encouraging. However, applying principal component analysis to reduce the large feature vector has only decreased the execution time without enhancing the classification performance.

Category	Sub-Category	Cited method	Short description	Pros	Cons
Hand-crafted	Classical	[78]	μLBP+SVM	-Interpretable features -No data-augmentation -Few parameters to set -Small feature size -Robust to changes	-Low performance -Noise sensitivity -Difficult classifier choice -Heavy step of pre-processing
		[65]	GLCM+SVM		
		[28]	MSER detector		
		[147]	Hough transform		
		[109]	First order statistics		
		[103]	Haralick's features+KELM		
		[81]	Texture&shape+ bagged trees		
	Curvlet-based	[26]	Curvlet moments+KNN	-Reliable tissue representation	-Curse of dimensionality -Limited performance
		[102]	Curvlet coefficients		
		[71]	Curvlet+GLCM		
		[72]	Curvlet+regional features		
	Nature-inspired optimization	[79]	Otsu+Wind driven	-High accuracy rate -Effective feature selection mechanisms	-Heavy training step -Heavy parameter setting step
		[104]	Forrest optimization		
		[136]	Bee colony & Whale optimization		
	Uncertainty-based	[22]	Probability-based pervasive information features	-Robust to noise	-Heavy parameter setting step
		[52]	Fuzzy Logic+ Probabilistic NN		
Deep learning	Models from Scratch	[18]	9-layered CNN	-No fine tuning -High accuracy rate -No feature selection -Fully automated	-pre-processing needed -Data-augmentation -Time consuming
		[118]	CNN based BC-DROID		
		[125]	Deep encoder-decoder CNN		
		[85]	4-layered CNN: CNN-4d		
		[33]	CNN model		
	Transfer learning	[47]	Fine-tuned VGG-16	-High accuracy rate -No feature extraction	-Heavy fine-tuning -Overfitting -Model choice
		[36]	Fine-tuned Inception-V3 &ResNet50		
	Deep features+ classical classifier	[121]	CNN features+SVM	-High accuracy rate	-Large feature vector -Heavy feature selection

Table 4.1: Summary of the presented state-of-the-art methods.

For more readability, all the presented state-of-the-art methods are summarized in Table 4.1. A taxonomy to classify the reported methods as well as the pros and cons of each sub-category are presented in this Table. Overall, breast CAD systems based on hand-driven features have achieved acceptable accuracies for abnormality detection but still lack of precision when it comes to malignancy detection. Indeed, capturing discriminating information locally is necessary but not sufficient to construct robust features. The way the local information is aggregated, is crucial to retain the discriminative properties. Most of the methods use simple concatenation of local data upon different patches to construct the final feature, what can lead to the lost of useful local information. Moreover, efficiency of these techniques is based upon several parameters (*i.e.* thresholds, window sizes, numbers of cells...), which need to be manually fine-tuned. On the other side, breast CAD systems based on deep learning features have proven to be more accurate for malignancy detection but need huge amount of labeled data to perform training. Given that the majority of the publicly available datasets are not large enough to preform efficient learning of deep learning models, most of them suffer from overfitting problems even after having recourse to data augmentation and/or transfer learning. To tackle all these problems, we propose herein to handle the problem of aggregating local texture properties automatically. But, differently from the deep learning-based models, which need a large scale datasets to perform training, we propose to learn a genetic programming-based descriptor that generates automatically robust features while using small number of training instances.

4.3 Motivation and Contributions

In this chapter, we propose a non-Data-Hungry and Fully-Automated approach to Diagnosis Breast Cancer from mammographic images. The main contribution of the suggested method lies in the automation of the feature extraction step while using few training data. The same pipeline used to classify ROIs is also used for the CBMR with the only difference that for the latter, the similar ROIs are retrieved without giving an explicit decision. In this section, the overall process for malignant vs. benign mammogram ROIs classification/retrieval is detailed. It is worth noting that the same process followed to classify/retrieve the malignant vs. benign ROIs is exactly reproduced within the context of normal vs. abnormal ROIs. Figure 4.2 shows the different steps of this process. After splitting ROIs into training and test sets, the overall process consists of two main phases: an offline phase and an online one. In the offline phase, training ROIs are used to learn the adopted descriptor which is a Genetic Programming (GP)-based program that takes as input an ROI and generates the corresponding feature vector. The learned descriptor is used to generate a Knowledge Base (KB) which consists of triplets (ROI_i, L_i, X^i) (L_i and X^i correspond to the label and the generated feature vector for ROI_i, respectively) as shown in Figure 4.3. In the online phase, unseen ROIs are fed to the GP descriptor to generate the corresponding feature vectors. Finally, the k most similar ROIs are identified within the knowledge base using a distance measure between the feature vectors. For the CBMR case, the k most similar ROIs are simply returned and the classification of the breast mass (malignancy vs. benignity) in the input ROI is implicitly deduced from the labels L_i of the k ROIs retrieved from the knowledge base. If the majority of retrieved ROIs are benign then the mass is recognized as benign and vice versa. The main component of the proposed method is the GP-based descriptor that transforms each ROI image into a corresponding feature vector. In the following sub-sections, the overall process for generating the proposed descriptor is detailed. First, the suggested method for representing ROI texture locally using statistics on distribution of magnitude differences is presented. Then, the structure of the proposed GP program, which will act as a descriptor by transforming the local magnitude difference distributions into a rotation-invariant feature vector, is described. Finally, the genetic process for optimizing the descriptor as well as the defined fitness function are discussed.

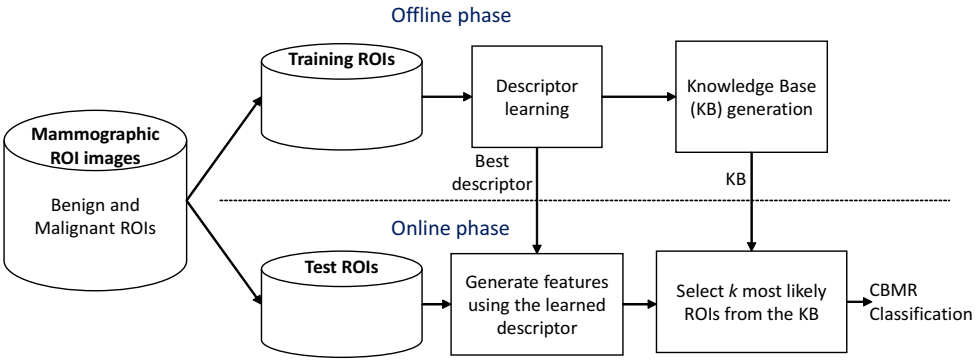


Figure 4.2: Flowchart of the proposed approach for breast cancer diagnosis from mammogram ROIs.

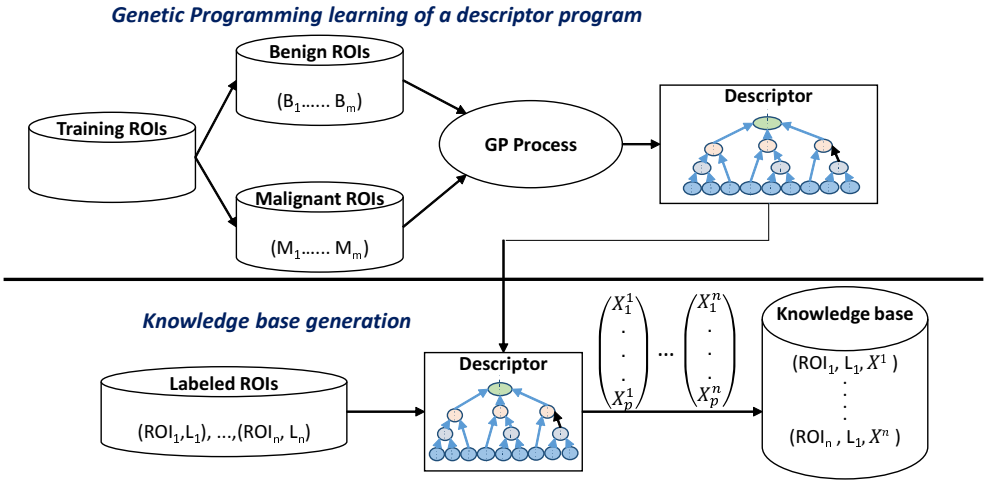


Figure 4.3: Overview of the offline phase for descriptor learning and knowledge base generation.

The suggested descriptor generates robust and discriminative features. Indeed, as illustrated in Figure 4.4, a population of descriptors are tested for their ability to satisfy the Fisher separation criterion when producing features from the training data. The quality of the features generated by the best descriptor of the current population is assessed based on a fitness measure. If the quality of the features is good enough, the best descriptor is selected otherwise an iteration of the genetic process is ran again in order to evolve the population of the descriptors. The selected descriptor at the end of this process is used to generate features for the test data to be fed to a supervised classifier in order to make a diagnosis decision. The task of the supervised classifier is easier since the quality of the generated features is guaranteed unlike features generated by classical descriptors, what allows improving the classification accuracy.

4.4 Local ROI texture representation

We define the texture T in a local neighborhood of an ROI using the joint distribution t of gray levels of $p+1$ pixels as follows (4.1):

$$T = t(g_c, g_0, \ldots, g_{p-1}), \tag{4.1}$$

where, g_c corresponds to the gray level of the central pixel within the neighborhood and g_k

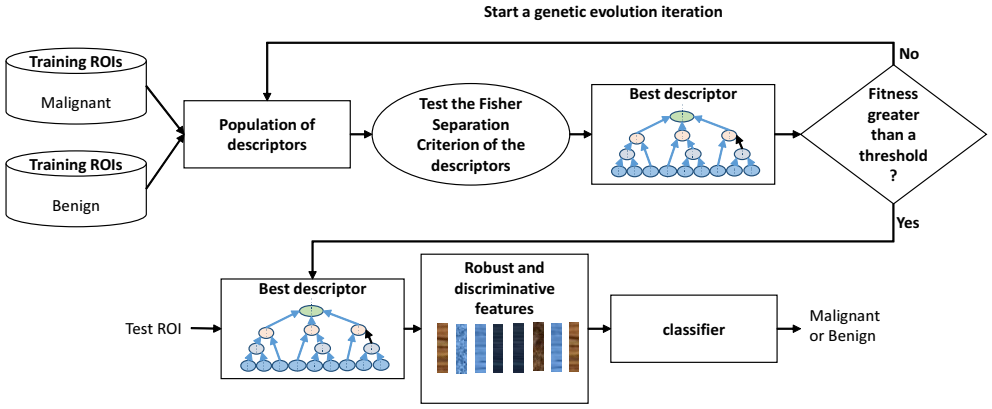


Figure 4.4: Genetic process for generating discriminative features that enhance the classification accuracy.

$(k = 0, \ldots, p-1)$ are the gray levels of its p neighbors (Figure 4.5). As in [112], the joint gray level distribution is approximated by the joint difference of magnitude distribution (4.2) in order to ensure gray scale invariance.

$$\begin{cases} T \approx t(d_0, \ldots, d_{p-1}), \\ \text{where } d_k = g_k - g_c, \ (k = 0, \ldots, p-1). \end{cases} \tag{4.2}$$

This representation is very discriminative since it captures variation of texture in each pixel neighborhood. The signs and the magnitudes of the differences tell a lot about the texture nature (constant, spot, edge, line...). Most of LBP variants transform this distribution to a binary string based on the signs of the d_k (0 if negative and 1 otherwise). Consequently, magnitude information is lost and only signs are considered. The main reason behind the binarization step is to reduce the number of possible patterns from 256^p to 2^p. Performing this step makes it possible to represent the global image with a 2^p-dimensional histogram feature or even less with other LBP variants, such as uniform LBP where the feature vector is of dimension $p(p-1)+3$. Despite the loss of magnitude information, LBP-like patterns have shown great robustness in classical texture classification problems. This is because classical texture classes vary significantly in uniformity, edge dispersion and patterns. However, for the mammogram ROIs, the texture is almost the same with a slight difference in density between the malignant/benign and normal/abnormal tissues, which is barely visible to the non-expert naked eye. Therefore, the magnitude loss in the LBP-based features can reduce considerably the classification accuracy but still necessary to reduce the pattern space size. Mapping the local texture distribution, while keeping the signs and the magnitudes into a fixed-size histogram feature, is also possible using a descriptor program. A descriptor transforms the local texture distribution in some keypoints to votes into a histogram feature. The direct mapping of the d_k $(k = 0, \ldots, k = p-1)$ signed magnitude differences using different operators (arithmetic, trigonometric...) is an alternative that does not warranty the rotation invariance since they depend on the initial orders assigned to the neighbor pixels. To overcome this problem, we consider order-independent statistics to characterize the magnitude difference distribution, namely min, max, $mean$, standard deviation ($stdev$), the Mean Absolute Deviation (MAD), the Root Mean Square (RMS), the Skewness (γ_1), the Kurtosis (β_2) and the Number of Changes (NOC) of the magnitude signs. The min, max, $stdev$ and $mean$ correspond to the minimum, maximum, standard deviation of the $(d_0, \ldots, d_{p-1})$ distribution. The NOC refers to the number of sign transitions (from + to − and inversely) circularly in the distribution (4.3). As shown in the example illustrated in Figure 4.5, the number of sign changes is equal to 4. The NOC value

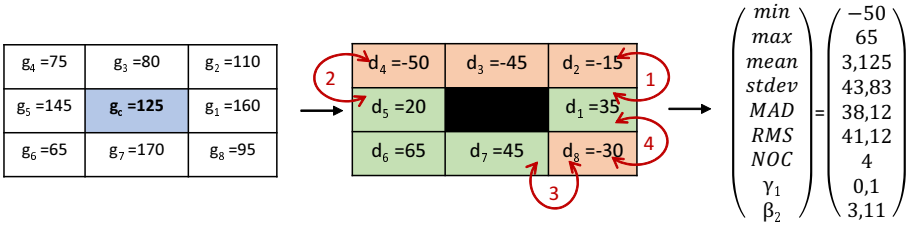


Figure 4.5: Statistics extraction from a 3×3 window.

captures information about the uniformity of the local texture. The MAD is the sum of the absolute differences between element values and the mean, divided by the sample size (p) as given by (4.4). The MAD value characterizes the variation of the magnitude differences around the mean. The RMS (4.5) describes the magnitude of the distribution elements, and the Skewness (4.6) describes how far to the left or to the right a distribution is distorted from a symmetrical bell curve. A distribution with a long left tail is left-skewed, or negatively-skewed; and a distribution with a long right tail is right-skewed, or positively-skewed. The Kurtosis (4.7) describes the extremeness of the tails of a distribution and is an indicator of data outliers. High kurtosis means that a data set has tail data that is more extreme than a normal distribution, and low kurtosis means that the tail data is less extreme than a normal distribution. Figure 4.5 illustrates an example of local texture distribution, the corresponding magnitude difference distribution and the set of statistics extracted from it. It is worth noting that all the statistics used to characterize the magnitude difference distribution are of order lower than 4. This is because those with order higher than 4 are computationally unstable when used for image analysis [88].

$$NOC = |\mathbb{1}_{\mathbb{R}^+}(d_{p-1}) - \mathbb{1}_{\mathbb{R}^+}(d_0)| + \sum_{i=1}^{p-1} |\mathbb{1}_{\mathbb{R}^+}(d_i) - \mathbb{1}_{\mathbb{R}^+}(d_{i-1})|. \tag{4.3}$$

where, $\mathbb{1}_S$ denotes the characteristic function of a subset S.

$$MAD = \frac{1}{p} \sum_{i=0}^{p-1} |d_i - mean|. \tag{4.4}$$

$$RMS = \sqrt{\frac{\sum_{i=0}^{p-1} d_i^2}{p}}. \tag{4.5}$$

$$\gamma_1 = \frac{p}{(p-1)(p-2)} \sum_{i=0}^{p-1} \left(\frac{d_i - mean}{stdev}\right)^3. \tag{4.6}$$

$$\beta_2 = \frac{p(p+1)}{(p-1)(p-2)(p-3)} \sum_{i=0}^{p-1} \left(\frac{d_i - mean}{stdev}\right)^4. \tag{4.7}$$

4.4.1 Program representation

As mentioned in the previous sub-section, to keep magnitude and sign informations, we have to define a program structure that allows to transform the set of statistics calculated for the local magnitude distribution, on a set of ROI pixels, to a feature vector representing the whole ROI image. To achieve this, a tree structure program is defined. In fact, a program tree is made up of a root node, a number of internal nodes and leaf nodes. An example of a program is depicted in Figure

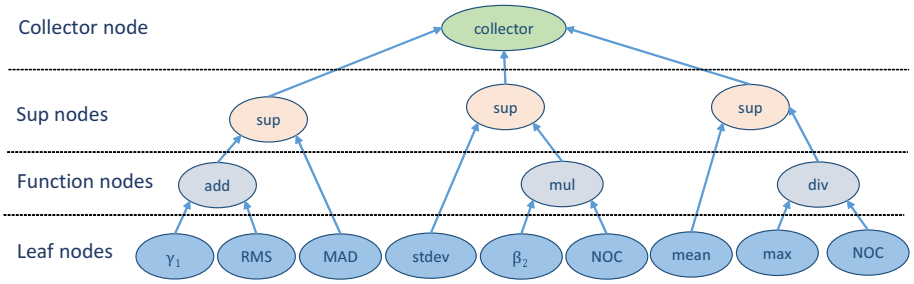


Figure 4.6: Example of a program tree structure for a 3-bit code descriptor.

4.6. The terminal set (leaf nodes) consists of nodes which are chosen among the set of statistics $\{min, max, stdev, mean, MAD, RMS, NOC, \gamma_1, \beta_2\}$ as previously detailed. The non-terminal set is composed of *root node*, *root children nodes* and *function nodes*. A *function node* is chosen within a set of arithmetic operators $\{add, sub, mul, div\}$, which have the same input and output type. An exception for the division operator, it returns zero if the denominator is zero. The *root node* is the *collector node* and it is responsible for collecting the results from the children nodes and constructing the feature vector. This node will be detailed later when explaining how the feature vector is extracted. The root children are *sup* nodes, and their number will specify the size of the feature vector. The *sup* node (Figure 4.6) is a binary operator that returns 1 if the left child is greater than the right child and 0 otherwise.

4.4.2 Feature vector extraction

The non-terminal nodes are evaluated starting from the leaf nodes up to the root children by applying the corresponding operator to the child nodes. The collector node produces a binary code from the root children (*sup* nodes) as shown in Figure 4.7. The length of the binary code is specified by the number of the *sup* nodes. In the remaining of this paper, *code length* (cl) denotes the number of *sup* nodes of an individual. An individual with cl *sup* nodes, generates a $cl - bit$ binary code. This binary code is used to construct a 2^{cl} feature vector. The decimal equivalent of the generated binary code will indicate the bin of the feature vector for which the vote will be allocated. Figure 4.7 illustrates how a tree-based individual, where the value of *code length is equal to* 3, transforms the set of statistics, calculated for a given local texture distribution, to a vote within the global histogram feature. Each leaf node of the individual tree refers to an element of the set of operators $\{min, max, stdev, mean, MAD, RMS, NOC, \gamma_1, \beta_2\}$, and the value of a parent node is defined in a bottom-up manner from the children nodes with the corresponding operator. Each *sup* node returns a binary value (1 if the left child value is greater than the right one and 0 otherwise), and the collector node (root) collects the final binary code (= 110 in the example of Figure 4.7) and converts it to the corresponding decimal number (= 6 in the example of Figure 4.7). The decimal number obtained represents the bin into the final histogram feature to which the vote will be allocated. The 3-*code length* descriptor in this example generates an 8-dimensional feature vector (= 2^3). The genetic encoding of the descriptors and the feature extraction being detailed, it remains to define the fitness function that will allow the GP algorithm to elect relevant descriptors.

4.4.3 Fitness function

In genetic programming-based classification, the rate of correctly classified instances is, generally, taken as a fitness measure. In our case, evolved individuals are descriptors and not classifiers. A good descriptor is the one that generates the best features to be fed to a classifier. To enhance

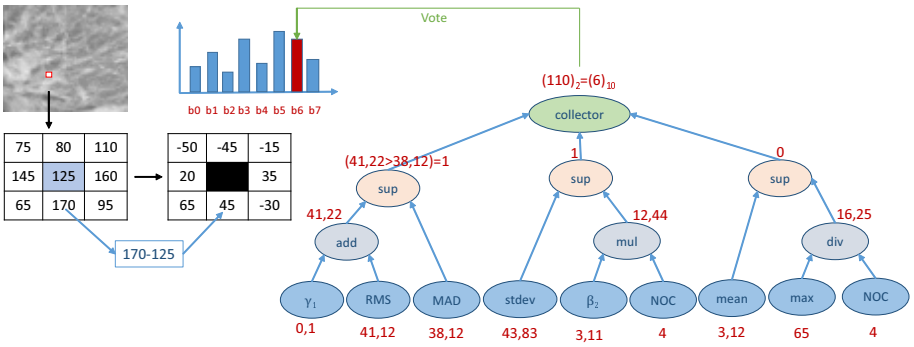


Figure 4.7: Feature vector extraction: a 3-bit descriptor transforming the statistics on a local texture distribution to a vote in an 8-bit histogram feature.

the classification task, features must be close in the case of instances of the same class and very discriminating when it comes to different classes. The proposed fitness measure (4.8) takes into account the homogeneity of features inside each class and their discrimination power when dealing with different classes.

$$fitness = 1 - \frac{log(2)}{log(1 + e^{D_c/H_c})}, \tag{4.8}$$

where, H_c (4.9) is the homogeneity coefficient that describes how strongly feature vectors of instances of the same class resemble each other, and D_c (4.10) is the discrimination coefficient that describes how strongly feature vectors of instances of different classes are distant from each other. Indeed, the H_c (*resp.* D_c) coefficient illustrates the average intra-class (*resp.* inter-class) similarity measure between training features. Both H_c and D_c coefficients range from 0 to 1. Good intra-class features' homogeneity corresponds to H_c values close to 0, and the discriminating power of a descriptor grows for D_c values nearby 1. Thus, the best individuals are those with higher D_c/H_c ratios (Figure 4.8).

$$H_c = \frac{1}{m(m-1)}\left(\sum_{i<j}\chi^2(X_i^B, X_j^B) + \sum_{i<j}\chi^2(X_i^M, X_j^M)\right), \tag{4.9}$$

$$D_c = \frac{1}{m^2}\sum_{i,j}\chi^2(X_i^B, X_j^M), \tag{4.10}$$

where, m is the number of training instances per class, X_i^B (resp. X_i^M) is the normalized feature vector of the i^{th} ($i \in \{1, \ldots, m\}$) training instance of the benign class (resp. malign class) , and $\chi^2(U,V)$ (4.11) is the Chi-square distance ($\in [0, 1]$) between two normalized feature vectors of the same dimension n.

$$\chi^2(X,Y) = \frac{1}{2}\sum_{i=1}^{n}\frac{(X_i - Y_i)^2}{X_i + Y_i}. \tag{4.11}$$

The final feature vector is constructed using all the pixels of the input ROI image. Since the used ROIs are of size 128×128, a total number of 16384 votes constitutes the final vector. To find the trade-off between the discriminative power and the stability of the feature vector, an $8-bit$ program (with 8 sup nodes) is chosen to generate a $2^8 = 256$-bin histogram feature. The size of the code length ($cl = 8$) is chosen to warranty the stability of the resulting feature. Indeed, an average of 64 (=16384/256) entries per bin of the histogram is ensured, what represents a good trade-off between stability and discrimination power (ensured by a minimum of 10 entries according to [112]).

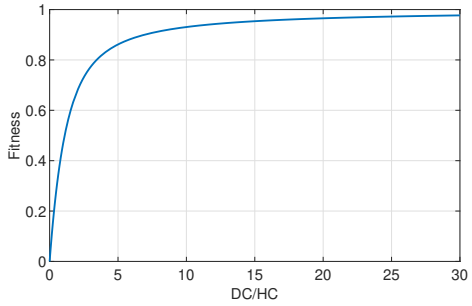


Figure 4.8: Fitness measure as a function of the $^{D_c}/_{H_c}$ ratio.

4.4.4 Genetic programming-based descriptor program generation

Genetic programming is used herein in order to optimize an initial population of randomly generated programs. Each random program has a tree-based structure as described in subsection 4.4.1. It is evaluated using the fitness function described in 4.4.3 while considering a training set that includes m malignant ROIs and m benign ones. The GP algorithm works in an iterative fashion, where at each iteration, a population of n descriptor programs is evolved as shown in Algorithm 2. The best program obtained over the generations will act as descriptor in order to generate the feature vectors for the input ROI images and to produce thereafter the knowledge base to assess the unseen data. We notice that using a GP-based descriptor is the key to avoid a data-hungry learning. Indeed, the principle of evolutionary techniques is adopted to generate a population of descriptors while assessing their ability to generate discriminative features. For binary class problems, it is possible to evolve a population using only small number of samples from each class. In fact, evolution is acting in a manner to elect descriptors that are able to extract discriminative features for the training samples. Running the evolution process, with randomly chosen subset of the training set for each generation, allows assessing the ability of the descriptors to generalize over new instances while avoiding overfitting.

Furthermore, the parameter setting of the genetic process for evolving a descriptor is summarized in Table 4.2. In fact, the "ramped half-and-half" method is applied in order to generate the initial population, such that the population size is set to 200 individuals. The tournament selection strategy with a tournament of size 7 is used to maintain the population diversity, and the crossover and mutation probabilities are set to 0.80 and 0.20, respectively. We adopt the "keep the best" mechanism to prevent the evolutionary process from degrading. Moreover, the tree depth of an evolved program is between 2 and 10 levels in order to avoid code bloating. To end with, the evolving process stops when the ideal individual is found: fitness value is equal to 1 or very close to the ideal (*e.g.* 10^{-6}), or the maximum number of generations is reached (*e.g.* 30).

4.5 Classification and ROI retrieval

In order to evaluate the ability of the GP-based descriptors to generate discriminative features allowing to distinguish between normal and abnormal tissues and between benign and malignant lesions, the *KNN* classifier is adopted. Indeed, the choice of *KNN* is motivated by its stability and simplicity compared to other well known classifier (random forest, neural networks...) [17]. In addition, the *KNN* classifier is the most used classifier within the framework of breast cancer diagnosis based on feature classification [126, 119]. Since the proposed descriptor was genetically designed based on the Chi-square distance as similarity measure between feature vectors, the same distance metric (4.11) is used to find the k nearest ROI neighbors of the input ROI. In fact, in order to predict the class label

Algorithm 2: Generate descriptor program using GP

Input: $(R_1^M,\ldots,R_m^M)$: malignant ROIs, $(R_1^B,\ldots,R_m^B)$: benign ROIs, MaxDepth, Code_length.

Output: Descriptor Program.

begin

- **for** $i = 1$ **to** n **do**
  - $P_i \longleftarrow GenerateRandomProgram(MaxDepth, Code\_length)$
- Current_pop$\longleftarrow \{P_1,\ldots,P_n\}$
- **while** *stop-criterion is not reached* **do**
  - **for** $P_i \in$ *Current_pop* **do**
    - **for** $j = 1$ **to** m **do**
      - $X_j^M \longleftarrow$generate_feature(P_i, R_j^M)
      - $X_j^B \longleftarrow$generate_feature(P_i, R_j^B)
      - `/* generate_feature is described in Figure 4.7 */`
    - `/* calculate fitness(`P_i`) using Eq.4.8 */`
    - **if** *fitness*(P_i)>*fitness*(P_{best}) **then**
      - $P_{best} \longleftarrow P_i$
  - Current_pop$\longleftarrow$evolve(Current_pop)
- return(P_{best})

Parameter	**Value**	**Parameter**	**Value**
Crossover rate	*0.80*	Generations	*30*
Mutation rate	*0.20*	Population size	*200*
Elitism	*keep the best*	Initial population	*Ramped half-and-half*
Tree min depth	*2*	Selection type	*Tournament*
Tree max depth	*10*	Tournament size	*7*

Table 4.2: Parameter setting of the genetic process.

Dataset	Nb. ROIs	Nb. normal ROIs	Abnormal ROIs			Size
			Total Nb.	benign ROIs	Nb. malignant ROIs	
DDSM	11218	9215	2003	888	1115	128×128
MIAS	322	207	115	63	52	128×128

Table 4.3: A summary of the used benchmark mammographic ROI datasets.

C_R of a sample ROI R, the majority voting strategy is applied (4.12).

$$C_R = \underset{C}{\operatorname{argmax}} \sum_{i=1}^{K} \delta(L_i, C), \tag{4.12}$$

where, L_i is the label of the i^{th} neighbor of the sample R and $\delta(L_i, C) = 1$ if $L_i = C$ and 0 otherwise.

4.6 Results and discussion

The proposed method is evaluated along with two datasets as summarized in Table 4.3.

The proposed method is assessed within the framework of retrieving mammogram ROIs based on their content. For the evaluation of mammograms retrieval, only the DDSM dataset has been considered since it contains a large set of ROIs. For content-based retrieval experiments, the 10 most relevant ROI instances are considered since radiologists pay most attention on the top ten retrieved images [27].

The proposed CBMR is firstly evaluated qualitatively. Figure 4.9 illustrates the ROI retrieval for normal and abnormal ROI samples. All the retrieved ROIs belong to the same classes of the input ROIs. Indeed, normal and abnormal tissues present different textures and it is relatively easy to differentiate between them. However, for the malignant vs. benign ROI case, the problem is quite more difficult since some inter-class breast lesions present similar texture in mammographic images. As shown in Figure 4.10, the 5^{th} and the 8^{th} retrieved instances for the input benign ROI are malignant lesions that were miss-classified within the 10 ROIs. Similarly, within the 10 retrieved ROIs for the input malignant sample, 3 instances (with red frame) were miss-classified. These instances correspond to the benign class but present similar texture characteristics as the malignant input lesion. In both benign and malignant cases, the majority of the retrieved instances are correctly classified, and for the example illustrated in Figure 4.10, the first miss-retrieved ROI appears in the 4^{th} rank.

The proposed method is then assessed for the classification problem. For fair comparison, we ran the experiments using the same data for training and testing while using the same validation strategy. For this, the test dataset is randomly partitioned into 5 disjoint folds of equal size. For each fold, the accuracy is estimated through the classification of the samples of this fold based on the other four folds. This process is repeated twice. Besides, in order to evaluate the statistical significance of performance differences between the suggested method and the compared ones, the paired t-test was performed on the results of the 2×5-fold cross validation. At the significance of level of 5%, we consider that the mean accuracies of the proposed method and that of the compared methods are equal, otherwise the difference is significant. All the compared methods are used with optimal parameters as mentioned by the authors in the original papers. As illustrated in Table 4.4, the suggested method, statistically outperformed all the compared methods, except for the methods in [136] and [28] where the p-values are 2.69% and 4.88%, respectively. Thus, when tested and trained using the same data, the proposed method performs better than all the compared methods. Indeed, incorporating feature selection and

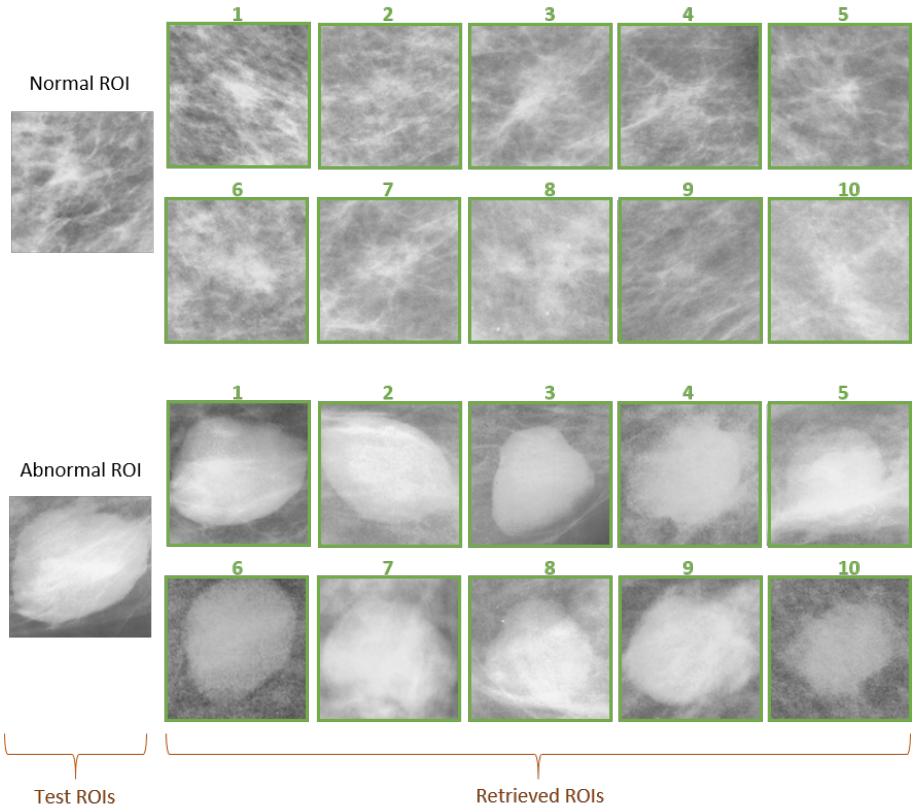


Figure 4.9: The 10 first normal and abnormal ROIs retrieved (rank #1 marks the closest ROI and rank #10 marks the most distant ROI).

fusion in the genetic non-data-hungry training step, for generating a context-aware descriptor, makes the classification task easier using a restricted set of prominent features. This allows to jointly improve the computational complexity and the performance of the proposed method compared to handcrafted methods and deep learning-based methods.

4.7 Discussion

This study deals with the problem of automating feature extraction for breast cancer diagnosis. A genetic programming-based descriptor is learned in order to capture discriminative information locally before generating a global feature. The suggested method is convenient for the classification as well as for CBMR issues. Indeed, all the ROIs used for the training with the corresponding labels and feature vectors are incorporated into a knowledge base during the learning process. Classification and CBMR are performed upon feature distance-based search over this knowledge base. The quality of the extracted features allows to perform accurate CBMR and classification of mammographic images. In fact, the proposed method has significantly outperformed all the compared baseline descriptors used in texture classification for both malignancy and abnormality detection. All these descriptors are handcrafted and have recorded good accuracies when dealing with usual texture classification problems. However, these accuracies have dropped significantly for the breast cancer classification problem. Indeed, analysing breast cancer tissue to separate between malignant and benign lesions, or even between normal and abnormal tissues, is a more difficult problem than classical texture classification. Dealing with slightly different inter-class texture makes feature extraction a more rigorous task that cannot be easily performed manually. Moreover, the baseline descriptors, such as LBP and GLCM,

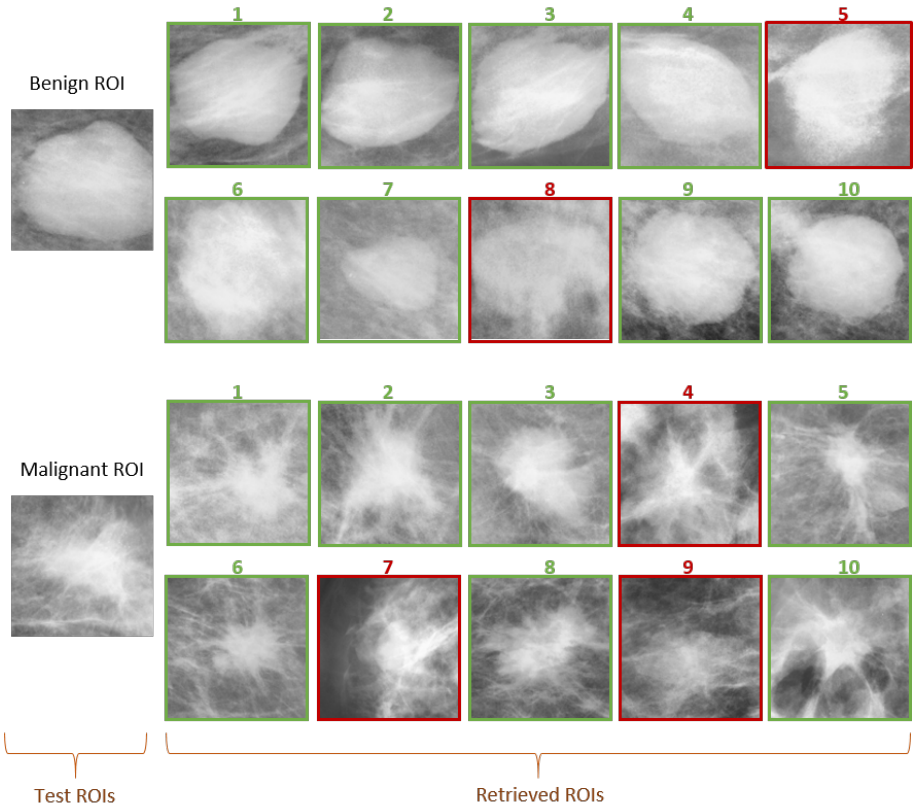


Figure 4.10: The 10 first malignant and benign ROIs retrieved (rank #1 marks the closest ROI and rank #10 marks the most distant ROI).

capture local information independently from the context of the classification. Thus, automating the feature extraction task will fit the extracted characteristics to the domain context. The suggested method has also outperformed all the methods based on features that were specially hand-crafted for breast cancer diagnosis when using unbiased LOOCV and 2×5-fold protocols. Indeed, the suggested descriptor is learned to extract features that capture local information so as to separate malignant (*resp.* abnormal) lesions from benign (*resp.* normal) tissues. It was designed genetically using a fitness function based on the Fisher separation criteria, while aggregating automatically the local texture features into a global one using arithmetic operators thanks to its tree based-structure. This aggregation mechanism is conducted by the evolutionary process in order to keep class-discriminative information in the global feature, unlike the classical concatenation mechanism used by several hand-driven methods. Furthermore, with regards to deep learning and transfer learning-based methods, which perform automated feature extraction, the suggested method is very competitive. However, most deep features are based on convolutional filters whereas the proposed local representation is based on robust statistics describing faithfully the local texture distribution. Moreover, the aggregation of the local representation is driven by a class-discriminative fitness function which preserves the class discriminating power within the global feature. All these aspects make the performance of the suggested method competitive with deep methods while performing better generalization capability and less overfitting. Indeed, the border between achieving good performance and overfitting is very narrow for deep learning-based methods. By dint of learning, they lose the ability to generalize outside the training data. In our case, we deal with this problem by using small number of training instances and by randomly changing a different subset of the training set through generations of the evolutionary process. Thus, the suggested method performs the best performance over the compared deep learning

Method	Partition 1					Partition 2					Overall
	Fold 1	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**Fold 1**	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	
[**121**]	89.85	90.14	89.78	91.45	91.20	90.51	90.27	90.07	90.78	90.45	90.45 ± 0.55
[**79**]	75.94	74.45	76.63	75.87	76.14	76.14	74.92	75.18	74.60	76.88	75.67 ± 0.84
[**104**]	80.96	81.17	82.14	81.56	80.90	81.61	81.3	82.07	83.78	80.23	81.57 ± 0.96
[**33**]	90.48	91.15	89.15	90.70	88.98	89.64	90.15	90.15	90.27	89.76	90.04 ± 0.67
[**28**]	90.95	89.51	91.13	89.14	90.92	91.12	91.43	90.3	90.14	91.21	90.58 ± 0.78
[**22**]	90.12	91.47	89.21	90.58	88.14	91.17	90.78	90.13	88.14	87.17	89.69 ± 1.46
[**81**]	88.95	89.78	88.14	87.14	88.12	89.98	87.18	88.56	89.12	88.73	88.64 ± 0.94
[**136**]	92.14	93.30	94.15	92.45	93.42	92.78	93.21	91.45	92.45	92.38	92.77 ± 0.77
Proposed	94.71	96.73	95.84	95.38	94.57	95.16	95.49	95.18	95.47	96.10	**95.46**±0.64

Table 4.4: Performance comparison against recent relevant state-of-the-art methods using the *2×5-fold* protocol on the DDSM dataset for abnormality detection.

methods on the MIAS dataset, which is relatively a small dataset (a total of 322 instances), without the need for data augmentation. This highlights another potential advantage of the proposed method, over deep learning, which is the "Non-Data-Hungry" aspect. In fact, bio-inspired approaches in general mimics the human logic for the learning mechanism. As human, we do not need to overwhelm the brain with training instances to have the ability to differentiate between different classes or categories. We just need to guide the learning process to detect discriminative characteristics over few samples of each class. Technically, this guiding role is played by the fitness function which makes it possible to orient the elitism towards discrimination between classes through small number of instances. To conclude, the proposed method performs feature selection and fusion automatically unlike non-deep learning-based methods. Most of these methods perform heavy pre-processing steps and must deal with feature selection manually or using feature reduction mechanisms such as principal component analysis. Non-automatic methods also fail to aggregate the local information into a global feature to characterize the whole region of interest because they use simple fusion techniques that do not necessary keep discriminative local information. The suggested method deals with the feature selection and fusion automatically thanks to the tree structure of the proposed descriptor. Features are selected using the evolutionary process and fused using the best combination of arithmetic operators that keep discriminative local information. For all these reasons, the proposed method performs well without having recourse to segmentation or any pre-processing step and gives better results compared to non-automatic methods as demonstrated in experimental results. The findings of the suggested framework demonstrate that fully automated and accurate image classification does not always mean data-hungry and that soft computing techniques offer an interesting alternative to deep learning for domains that lack data such as medical image analysis. It is worth noting that the suggested method has scored satisfying results using ROIs without region segmentation of suspicious lesions and has outperformed all the compared methods that performed pre-processing steps to separate breast abnormal masses. This finding supports the thesis of some research works on breast cancer stating that the texture at the border of the suspicious lesion is very important for the diagnosis [69]. The performance of the proposed method has been validated using the MIAS and the DDSM datasets. These labelled datasets are the largest publicly available collection of quality controlled mammographic images while including a representative collection of all important diagnostic categories in the realm of breast cancer diagnosis. Since the ground truth for all the cases within the used datasets was expert radiologist consensus, it is obvious that the genetic-programming-based features should be correlated with diagnosis and/or expert's observation. The quantitative evaluation demonstrates the ability of the proposed method to extract discriminative features for content-based image retrieval as well as for abnormality/malignancy

classification, given the true labels that are vetted by recognized expert mammography radiologists.

4.8 Conclusion

In this work, a fully automated method for local feature extraction and global feature generation for mammogram ROIs is proposed. To achieve this end, an LBP-like local representation is proposed, and a genetic programming-based descriptor is designed to transform local features into a global one. The evolutionary process is based upon a fitness function that guarantees the discriminative power of the descriptor while using small training instances. The suggested method has given encouraging results when applied for both content-based retrieval and classification problems. The finding of this research is extremely important for the breast cancer diagnosis since it simultaneously solve the problem of insufficient medical labelled data as well as the overfitting issue while being competitive with deep learning-based methods.

Chapter 5

Ongoing research and perspectives

I have presented throughout this manuscript most of my research work and the contributions that have emanated from it. I focused on three axes which are texture classification, facial expression recognition and breast cancer diagnosis. In the first axis, the major contribution concerns the presentation of a fully automated process that performs training with a limited number of labeled instances. Unlike classical machine learning techniques, the learning process in this work does not require hundreds of labeled images to perform an accurate classification. Similar to the human learning process, when it comes to learn the common characteristics for a group of images that differentiate it from the other groups, few instances from each group are sufficient. Indeed, when we try to teach a child how to differentiate between different animals, we do not flood his/her memory with examples of each animal, we do not also give him/her the features that characterize one animal from another and we do not teach him/her the way he/she should use these features to perform animal clustering. He/She only needs to see a few images of each animal to learn how to categorize them by learning the visual characteristics of each animal. Consequently, when feature extraction and aggregation are incorporated into the learning process, precise classification would be possible from a limited number of examples. This is what has been achieved in this axis since we teach the machine how to learn automatically a texture descriptor incorporating low-level feature extraction and global feature construction. Human ability to analyze texture locally, insensitively from scale and rotation, is also considered while defining the proposed local features. Thereby, the quality of the features given by the suggested GP descriptors makes it easy to a supervised classifier to perform accurate classification, even with a small number of training instances. Moreover, we should place the emphasis on the fully automatic aspect of the proposed method that does not need human expertise to select keypoints, to extract low-level feature or to construct a global feature. Indeed, texture features are used to detect wrinkles in human faces, which are important cues with geometric face deformations to capture human emotions from facial expressions. Common problems for texture classification, including local texture description, feature fusion, and small datasets are also issues for facial expression recognition and even medical image analysis, which brings us to the common thread between the three axes presented in this manuscript.

Similarly to the way we handled the aforementioned problems facing texture classification, we presented a 2D FER framework that performs fully automated feature fusion using small training data. Indeed, the suggested framework performs feature selection and fusion using binary programs evolved genetically. This ensures that the most discriminative features are adaptively selected for each pair of expressions. Actually, not all the features are significant to discriminate between all the facial expression classes. For example, the movements of the eyebrows can very well differentiate the *happiness* from the *surprise* but can mislead a classifier to select the wright expression between *surprise* and *fear*. This makes the proposed framework more accurate to classify facial expressions

than most of the approaches that perform feature selection globally. Besides, training the suggested 2D FER algorithm does not need large datasets or data augmentation, as it is the case for deep learning techniques. Indeed, to learn how to discriminate between two classes of facial emotions, the human brain does not need to be overwhelmed with instances from both classes. It only takes few instances for the brain to learn what features to select and how to fuse information from these features to perform accurate classification in unseen faces. In the same axis, we have explored the 3D/4D FER. An effective method for automated 3D/4D facial expression recognition based on Mesh-Local Binary Pattern Difference (mesh-LBPD) was proposed. In contrast to most of existing methods, the proposed mesh-LBPD is based on a unified set of geometric and appearance features of different facial regions. Indeed, multiple features are combined into a compact form using covariance matrices, namely $Cov-3D-LBP$. Then, the $Cov-3D-LBP$ atoms are represented as sparse data combinations. To that end, a Riemannian optimization objective for dictionary learning and sparse coding is used, in order to reduce the complexity of the problem, and the representation loss is characterized via an affine invariant Riemannian metric. The $Cov-3D-LBP$ descriptor has high discriminative power through blending multiple features. Moreover, it has good robustness derived from the following aspects. Firstly, it is robust to abrupt noise when the number of facets in a face region is large enough to form stable statistics. Secondly, the robustness can be enhanced if all selected elementary features have the consistent robustness (*i.e.* when only features insensitive to rotation are selected, the resulting $Cov-3D-LBP$ descriptor is robust to rotation changes). Thirdly, normalization usually leads to improved robustness by eliminating the affects coming from different variances of features.

Most of medical application fields suffer from difficulties to label data. Therefore, the studies presented in the first two axes motivated us to explore medical image analysis. Consequently, the idea of fully automated texture feature extraction and selection is extended to the problem of breast cancer diagnosis, which bring us to the third axis presented in this manuscript. Indeed, analysing local texture and generating features are two key issues for automatic cancer detection in mammographic images. Recent researches have shown that deep neural networks provide a promising alternative to hand-driven features which suffer from curse of dimensionality and low accuracy rates. However, large and balanced training data are foremost requirements for deep learning-based models and these data are not always available publicly. In the third axis of this manuscript, we proposed a fully-automated method for breast cancer diagnosis that performs training using small sets of data. the proposed method performs feature selection and fusion automatically unlike non-deep learning-based methods. Most of these methods perform heavy pre-processing steps and must deal with feature selection manually or using feature reduction mechanisms such as principal component analysis. Non-automatic methods also fail to aggregate the local information into a global feature to characterize the whole region of interest because they use simple fusion techniques that do not necessary keep discriminative local information. The suggested method deals with the feature selection and fusion automatically thanks to the tree structure of the proposed descriptor. Local features are selected using the evolutionary process and fused using the best combination of arithmetic operators that keep discriminative local information. The proposed breast cancer diagnosis scheme performs well without having recourse to segmentation or any pre-processing step and gives better results compared to non-automatic methods. The findings of the suggested framework demonstrate that fully automated and accurate image classification does not always mean data-hungry and that soft computing techniques offer an interesting alternative to deep learning for domains that lack data such as medical image analysis.

In the same direction and in addition to the works already considered, my current and future works aim at the following points. In the context of texture classification, we are investigating the performance of the suggested HL-GP descriptor with different standard classifiers for multi-class clas-

sification. This would help to assess the robustness of the produced features. We are also studying the possibility of using other algorithms that could also be adapted directly as GP functions such as edge detection ones. In the context of FER, we are focusing on subject-dependent feature selection. This choice is motivated by the specificity of each human face, which makes the spontaneous facial emotion display differs from one subject to another. To address this problem, we suggest a face-based dynamic feature selection. The proposed selection mechanism provides a better understanding of the facial transformation during the emotion display. Moreover, the suggested selection method takes into consideration the subject's general facial structure, muscle movements, and head position. On another side, the swift progress in Deep Learning (DL) motivated us to explore DL techniques for facial expression recognition. In this context, we are investigating a multichannel convolutional neural network based on the quality and strengths of several well-known pre-trained DL models. Indeed, the complementarity of the features extracted from the studied models will be exploited to form a more robust feature vector. In the context of breast cancer diagnosis, we aim to validate the proposed method to other medical imaging applications involving classification and/or content-based retrieval. Moreover, medical data augmentation using generative adversial genetic programs, where two generative and discriminatory genetic programs compete with each other in order to produce new instances, seems to be an interesting topic that could be explored.

Bibliography

[1] T. Ahonen, A. Hadid, and M. Pietikainen. "Face Description with Local Binary Patterns: Application to Face Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.12 (2006), pp. 2037–2041.

[2] N. Aifanti, C. Papachristou, and A. Delopoulos. "The MUG facial expression database". In: *11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10*. 2010, pp. 1–4.

[3] Legaz-Aparicio Álvar-Ginés, Verdú-Monedero Rafael, and Engan Kjersti. "Noise robust and rotation invariant framework for texture analysis and classification". In: *Applied Mathematics and Computation* 335 (2018), pp. 124–132. ISSN: 0096-3003. DOI: https://doi.org/10.1016/j.amc.2018.04.018.

[4] Backes André Ricardo. "Upper and lower volumetric fractal descriptors for texture classification". In: *Pattern Recognition Letters* 92 (2017), pp. 9–16. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2017.03.020.

[5] Amany Abdel Aziz Arafa et al. "Computer-Aided Detection System for Breast Cancer Based on GMM and SVM". In: *Arab Journal of Nuclear Sciences and Applications* 52.2 (2019), pp. 142–150.

[6] Daniel Atkins, Kourosh Neshatian, and Mengjie Zhang. "A domain independent Genetic Programming approach to automatic feature extraction for image classification". In: *2011 IEEE Congress of Evolutionary Computation (CEC)*. IEEE, June 2011. DOI: 10.1109/cec.2011.5949624.

[7] H. Bejaoui, H. Ghazouani, and W. Barhoumi. "Fully Automated Facial Expression Recognition Using 3D Morphable Model and Mesh-Local Binary Pattern". In: *Advanced Concepts for Intelligent Vision Systems*. Cham: Springer International Publishing, 2017, pp. 39–50.

[8] Hela Bejaoui, Haythem Ghazouani, and Walid Barhoumi. "Fully Automated Facial Expression Recognition Using 3D Morphable Model and Mesh-Local Binary Pattern". In: Nov. 2017, pp. 39–50. ISBN: 978-3-319-70352-7. DOI: 10.1007/978-3-319-70353-4_4.

[9] Hela Bejaoui, Haythem Ghazouani, and Walid Barhoumi. "Sparse coding-based representation of LBP difference for 3D/4D facial expression recognition". In: *Multimedia Tools and Applications* 78 (Aug. 2019), pp. 22773–22796. DOI: 10.1007/s11042-019-7632-2.

[10] B. Ben Amor et al. "4-D facial expression recognition by learning geometric deformations". In: *IEEE T. Cybernetics* 44.12 (2014), pp. 2443–2457.

[11] Asa Ben-Hur and Jason Weston. “A User’s Guide to Support Vector Machines”. In: *Methods in molecular biology (Clifton, N.J.)* 609 (Jan. 2010), pp. 223–39. DOI: 10.1007/978-1-60327-241-4_13.

[12] Balas Benjamin J. “Texture synthesis and perception: Using computational models to study texture representations in the human visual system”. In: *Vision Research* 46.3 (2006), pp. 299–309. ISSN: 0042-6989. DOI: https://doi.org/10.1016/j.visres.2005.04.013.

[13] S. Berretti et al. “A Set of Selected SIFT Features for 3D Facial Expression Recognition”. In: *2010 20th International Conference on Pattern Recognition*. 2010, pp. 4125–4128.

[14] Alina Beygelzimer, John Langford, and Pradeep Ravikumar. “Multiclass classification with filter trees”. In: *Preprint, June* 2 (2007).

[15] Ying Bi, Mengjie Zhang, and Bing Xue. “Genetic programming for automatic global and local feature extraction to image classification”. In: *2018 IEEE Congress on Evolutionary Computation (CEC)*. 2018, pp. 1–8.

[16] Hadjer Boughanem, Haythem Ghazouani, and Walid Barhoumi. “Towards a deep neural method based on freezing layers for in-the-wild facial emotion recognition”. In: *2021 IEEE/ACS 18th International Conference on Computer Systems and Applications (AICCSA)*. 2021, pp. 1–8. DOI: 10.1109/AICCSA53542.2021.9686927.

[17] Leo Breiman. “Bagging Predictors”. In: *Machine Learning* 24.2 (1996), pp. 123–140.

[18] S. Charan, M. J. Khan, and K. Khurshid. “Breast cancer detection in mammograms using convolutional neural network”. In: *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*. 2018, pp. 1–5. DOI: 10.1109/ICOMET.2018.8346384.

[19] A. Cherian and S. Sra. “Riemannian Dictionary Learning and Sparse Coding for Positive Definite Matrices”. In: *IEEE Transactions on Neural Networks and Learning Systems* 28.12 (2017), pp. 2859–2871.

[20] Mircea Cimpoi et al. “Describing textures in the wild”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 3606–3613.

[21] M Crosier and Lewis Griffin. “Using Basic Image Features for Texture Classification”. In: *Int J Comput Vision* 88 (July 2010), pp. 447–460. DOI: 10.1007/s11263-009-0315-0.

[22] Jyoti Dabass, M. Hanmandlu, and Rekha Vig. “Formulation of probability-based pervasive information set features and Hanman transform classifier for the categorization of mammograms”. In: *SN Applied Sciences* 3.6 (May 2021), p. 610.

[23] N. Dalal and B. Triggs. “Histograms of oriented gradients for human detection”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 1. 2005, pp. 886–893.

[24] N. Dalal and B. Triggs. “Histograms of Oriented Gradients for Human Detection”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. IEEE. DOI: 10.1109/cvpr.2005.177.

[25] Per-Erik Danielsson. “Euclidean distance mapping”. In: *Computer Graphics and Image Processing* 14.3 (Nov. 1980), pp. 227–248. DOI: 10.1016/0146-664x(80)90054-4.

[26] Sami Dhahbi, Walid Barhoumi, and Ezzeddine Zagrouba. “Breast cancer diagnosis in digitized mammograms using curvelet moments”. In: *Computers in Biology and Medicine* 64 (2015), pp. 79–90. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2015.06.012.

[27] Sami Dhahbi, Walid Barhoumi, and Ezzeddine Zagrouba. “Multi-view score fusion for content-based mammogram retrieval”. In: Dec. 2015. DOI: 10.1117/12.2228614.

[28] B. V. Divyashree and G. Hemantha Kumar. “Breast Cancer Mass Detection in Mammograms Using Gray Difference Weight and MSER Detector”. In: *SN Computer Science* 2.2 (Jan. 2021), p. 63.

[29] Jyotsna Dogra, Shruti Jain, and Meenakshi Sood. “Glioma extraction from MR images employing Gradient Based Kernel Selection Graph Cut technique”. In: *The Visual Computer* (2019), pp. 1–17. DOI: https://doi.org/10.1007/s00371-019-01698-3.

[30] P. Ekman. “Universals and cultural differences in facial expressions of emotion”. In: *In Nebraska Symposium on Motivation* (1972), pp. 207–283.

[31] Paul Ekman and Wallace V. Friesen. “Facial action coding system: a technique for the measurement of facial movement”. In: *Palo Alto, CA: Consulting Psychologists Press* (1978).

[32] Paul Ekman and Wallace V. Friesen. “Felt, false, and miserable smiles”. In: 6.4 (1982), pp. 238–252. DOI: 10.1007/bf00987191. URL: https://doi.org/10.1007/bf00987191.

[33] Enas El Houby and Nisreen Yassin. “Malignant and nonmalignant classification of breast lesions in mammograms using convolutional neural networks”. In: *Biomedical Signal Processing and Control* 70 (2021), p. 102954. ISSN: 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2021.102954.

[34] G. B. Ernesto, M. M. José, and R. Marcos. “Algorithm 813: SPG—Software for Convex-Constrained Optimization”. In: *ACM Transactions on Mathematical Software*. Vol. 27. 2001, pp. 340–349.

[35] P.G. Espejo, S. Ventura, and F. Herrera. “A Survey on the Application of Genetic Programming to Classification”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 40.2 (Mar. 2010), pp. 121–144. DOI: 10.1109/tsmcc.2009.2033566.

[36] Lenin G. Falconí, María Pérez, and Wilbert G. Aguilar. “Transfer Learning in Breast Mammogram Abnormalities Classification With Mobilenet and Nasnet”. In: *2019 International Conference on Systems, Signals and Image Processing (IWSSIP)*. 2019, pp. 109–114. DOI: 10.1109/IWSSIP.2019.8787295.

[37] Antonio Fernández, Marcos Álvarez Cid, and Francesco Bianconi. “Texture Description Through Histograms of Equivalent Patterns”. In: *Journal of Mathematical Imaging and Vision* 45 (Jan. 2013), pp. 76–102. DOI: 10.1007/s10851-012-0349-8.

[38] Haythem Ghazouani. “A genetic programming-based feature selection and fusion for facial expression recognition”. In: *Applied Soft Computing* 103 (2021), p. 107173. ISSN: 1568-4946.

[39] Haythem Ghazouani and Walid Barhoumi. “Genetic Programming-based Learning of Texture Classification Descriptors from Local Edge Signature”. In: *Expert Systems with Applications* 161 (June 2020), p. 113667. DOI: 10.1016/j.eswa.2020.113667.

[40] Haythem Ghazouani and Walid Barhoumi. “Genetic programming-based learning of texture classification descriptors from Local Edge Signature”. In: *Expert Systems with Applications* 161 (2020), p. 113667. ISSN: 0957-4174.

[41] Haythem Ghazouani and Walid Barhoumi. “Towards non-data-hungry and fully-automated diagnosis of breast cancer from mammographic images”. In: *Computers in Biology and Medicine* 139 (2021), p. 105011. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2021.105011. URL: https://www.sciencedirect.com/science/article/pii/S0010482521008052.

[42] Haythem Ghazouani, Walid Barhoumi, and Yosra Antit. “A Genetic Programming Method for Scale-Invariant Texture Classification”. In: *Proceedings of the 21st International Conference on Engineering Applications of Neural Networks (EANN)*. June 2020, pp. 605–616. DOI: 10.1007/978-3-030-48791-1_47.

[43] D. Ghimire and J. Lee. “Geometric feature-based facial expression recognition in image sequences using multi-class AdaBoost and support vector machines”. In: *Sensors*. Vol. 13. 2016, pp. 7714–7734.

[44] Deepak Ghimire et al. “Recognition of facial expressions based on salient geometric features and support vector machines”. In: *Multimedia Tools and Applications* 76 (2017), pp. 7921–7946. DOI: 10.1007/s11042-016-3428-9.

[45] Ivo Gonçalves and Sara Silva. “Balancing Learning and Overfitting in Genetic Programming with Interleaved Sampling of Training Data”. In: Apr. 2013, pp. 73–84. DOI: 10.1007/978-3-642-37207-0_7.

[46] B. Gong et al. “Automatic Facial Expression Recognition on a Single 3D Face by Exploring Shape Deformation”. In: *Proceedings of the 17th ACM International Conference on Multimedia*. MM ’09. 2009, pp. 569–572.

[47] S. Guan and M. Loew. “Breast Cancer Detection Using Transfer Learning in Convolutional Neural Networks”. In: *2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*. 2017, pp. 1–8. DOI: 10.1109/AIPR.2017.8457948.

[48] Zhenhua Guo, Lei Zhang, and David Zhang. “A Completed Modeling of Local Binary Pattern Operator for Texture Classification”. In: *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society* 19 (Mar. 2010), pp. 1657–63. DOI: 10.1109/TIP.2010.2044957.

[49] S. Hamit and D. Hasan. “Facial Expression Recognition Using 3D Facial Feature Distances”. In: *Image Analysis and Recognition*. Springer Berlin Heidelberg, 2007.

[50] H. Han et al. “Demographic estimation from face images: Human vs. machine performance”. In: *IEEE Transaction on Pattern Analysis and Machine Intelligence* 19 (2015), pp. 1148–1161.

[51] W. Hariri et al. “3D facial expression recognition using kernel methods on Riemannian manifold”. In: *Engineering Applications of Artificial Intelligence* 64 (2017), pp. 25–32. ISSN: 0952-1976.

[52] T. M. Hasan, S. D. Mohammed, and J. Waleed. “Development of breast cancer diagnosis system based on fuzzy logic and probabilistic neural network”. In: *Eastern-European Journal of Enterprise Technologies* 4.9 (106) (Aug. 2020), pp. 6–13. DOI: 10.15587/1729-4061.2020.202820.

[53] Mohamed Hazgui, Haythem Ghazouani, and Walid Barhoumi. “Evolutionary-based generation of rotation and scale invariant texture descriptors from SIFT keypoints”. In: *Evolving Systems* 12.3 (May 2021), pp. 583–590. ISSN: 1868-6486. DOI: 10.1007/s12530-021-09386-1.

[54] Mohamed Hazgui, Haythem Ghazouani, and Walid Barhoumi. “Genetic programming-based fusion of HOG and LBP features for fully automated texture classification”. In: *The Visual Computer* (Jan. 2021), In press. ISSN: 1432-2315. DOI: 10.1007/s00371-020-02028-8.

[55] M. He et al. “Analyses of the Differences between Posed and Spontaneous Facial Expressions”. In: *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*. 2013, pp. 79–84.

[56] X. Hong et al. “Combining LBP Difference and Feature Correlation for Texture Description”. In: *IEEE Transactions on Image Processing* 23.6 (2014), pp. 2557–2568.

[57] T. T. Htay and S. S. Maung. “Early Stage Breast Cancer Detection System using GLCM feature extraction and K-Nearest Neighbor (k-NN) on Mammography image”. In: *2018 18th International Symposium on Communications and Information Technologies (ISCIT)*. 2018, pp. 171–175. DOI: 10.1109/ISCIT.2018.8587920.

[58] Yuting Hu, Zhiling Long, and Ghassan Alregib. “Scale Selective Extended Local Binary Pattern for Texture Classification”. In: Mar. 2017, pp. 1413–1417. DOI: 10.1109/ICASSP.2017.7952389.

[59] Yibin Huang, Congying Qiu, and Kui Yuan. “Surface defect saliency of magnetic tile”. In: *The Visual Computer* 36.1 (2020), pp. 85–96. DOI: https://doi.org/10.1007/s00371-018-1588-5.

[60] L. Huibin, M. Jean-Marie, and L. Liming. “3D Facial Expression Recognition Based on Histograms of Surface Differential Quantities”. In: *Advanced Concepts for Intelligent Vision Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 483–494.

[61] Bayezid Islam, Firoz Mahmud, and Arfat Hossain. “High Performance Facial Expression Recognition System Using Facial Region Segmentation, Fusion of HOG & LBP Features and Multiclass SVM”. In: *2018 10th International Conference on Electrical and Computer Engineering (ICECE)*. IEEE. 2018, pp. 42–45. DOI: 10.1109/ICECE.2018.8636780.

[62] Bayezid Islam et al. “A Facial Region Segmentation Based Approach to Recognize Human Emotion Using Fusion of HOG & LBP Features and Artificial Neural Network”. In: *2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT)*. IEEE. 2018, pp. 642–646. DOI: 10.1109/CEEICT.2018.8628140.

[63] de Mesquita Jarbas Joaci and Backes André Ricardo. “ELM based signature for texture classification”. In: *Pattern Recognition* 51 (2016), pp. 395–401. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2015.09.014.

[64] Amal Jlassi et al. “Unsupervised Method based on Probabilistic Neural Network for the Segmentation of Corpus Callosum in MRI Scans”. In: *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SCITEPRESS - Science and Technology Publications, 2019. DOI: 10.5220/0007400205450552.

[65] G. R. Jothilakshmi and A. Raaza. “Effective detection of mass abnormalities and its classification using multi-SVM classifier with digital mammogram images”. In: *2017 International Conference on Computer, Communication and Signal Processing (ICCCSP)*. 2017, pp. 1–6. DOI: 10.1109/ICCCSP.2017.7944090.

[66] Amira Jouirou, Abir Baâzaoui, and Walid Barhoumi. “Multi-view information fusion in mammograms: A comprehensive overview”. In: *Information Fusion* 52 (2019), pp. 308–321. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2019.05.001. URL: http://www.sciencedirect.com/science/article/pii/S1566253518308091.

[67] Adrien Kaiser, Jose Alonso Ybanez Zepeda, and Tamy Boubekeur. “A survey of simple geometric primitives detection methods for captured 3d data”. In: *Computer Graphics Forum*. Vol. 38. 1. Wiley Online Library. 2019, pp. 167–196.

[68] Umasankar Kandaswamy, Stephanie A. C. Schuckers, and Donald Adjeroh. “Comparison of Texture Analysis Schemes Under Nonideal Conditions”. In: *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society* 20 (Dec. 2010), pp. 2260–75. DOI: 10.1109/TIP.2010.2101612.

[69] Anna N. Karahaliou et al. “Breast Cancer Diagnosis: Analyzing Texture of Tissue Surrounding Microcalcifications”. In: *IEEE Transactions on Information Technology in Biomedicine* 12.6 (2008), pp. 731–738. DOI: 10.1109/TITB.2008.920634.

[70] H. Karcher. “Riemannian center of mass and mollifier smoothing”. In: *Communications on Pure and Applied Mathematics* 30.5 (1977), pp. 509–541.

[71] R. Karthiga and K. Narasimhan. “Medical imaging technique using curvelet transform and machine learning for the automated diagnosis of breast cancer from thermal image”. In: *Pattern Analysis and Applications* 24.3 (Aug. 2021), pp. 981–991. ISSN: 1433-755X. DOI: 10.1007/s10044-021-00963-3.

[72] R. Karthiga, K. Narasimhan, and G. Usha. “Breast Cancer Diagnosis using Curvelet and Regional Features”. In: *2019 International Conference on Computer Communication and Informatics (ICCCI)*. 2019, pp. 1–5. DOI: 10.1109/ICCCI.2019.8821825.

[73] V. Kazemi and J. Sullivan. “One millisecond face alignment with an ensemble of regression trees”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1867–1874.

[74] Hanbay Kazim et al. “Continuous rotation invariant features for gradient-based texture classification”. In: *Computer Vision and Image Understanding* 132 (2015), pp. 87–101. ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2014.10.004.

[75] Ichrak Khoulqi and Najlae Idrissi. “Breast Cancer Image Segmentation and Classification”. In: *Proceedings of the 4th International Conference on Smart City Applications*. SCA ’19. Casablanca, Morocco: Association for Computing Machinery, 2019. ISBN: 9781450362894. DOI: 10.1145/3368756.3369039.

[76] M. Kopaczka, R. Kolk, and D. Merhof. “A fully annotated thermal face database and its application for thermal facial expression recognition”. In: *IEEE International Instrumentation and Measurement Technology Conference*. 2018, pp. 1–6.

[77] I. Kotsia and I. Pitas. “Facial Expression Recognition in Image Sequences Using Geometric Deformation Features and Support Vector Machines”. In: *IEEE Transactions on Image Processing* 16(1).1 (2007), pp. 172–187.

[78] P. Král and L. Lenc. “LBP features for breast cancer detection”. In: *2016 IEEE International Conference on Image Processing (ICIP)*. 2016, pp. 2643–2647. DOI: 10.1109/ICIP.2016.7532838.

[79] Romesh Laishram and Rinku Rabidas. “WDO optimized detection for mammographic masses and its diagnosis: A unified CAD system”. In: *Applied Soft Computing* 110 (2021), p. 107620. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2021.107620.

[80] Afshan Latif et al. “Content-based image retrieval and feature extraction: a comprehensive review”. In: *Mathematical Problems in Engineering* 2019 (2019).

[81] Ilhame Ait Lbachir, Imane Daoudi, and Saadia Tallal. “Automatic computer-aided diagnosis system for mass detection and classification in mammography”. In: *Multimedia Tools and Applications* 80.6 (Mar. 2021), pp. 9493–9525.

[82] V. L. Le, H. L. Tang, and T. S. Huang. “Expression recognition from 3D dynamic faces using robust spatio-temporal shape features”. In: *Automatic Face Gesture Recognition and Workshops IEEE International Conference* (2011), pp. 414–421.

[83] P. Lemaire et al. “Fully Automatic 3D Facial Expression Recognition Using a Region-based Approach”. In: *Proceedings of the 2011 Joint ACM Workshop on Human Gesture and Behavior Understanding*. J-HGBU ’11. 2011, pp. 53–58.

[84] Andrew Lensen et al. “Genetic Programming for Region Detection, Feature Extraction, Feature Construction and Classification in Image Data”. In: *Lecture Notes in Computer Science*. Springer International Publishing, 2016, pp. 51–67. DOI: 10.1007/978-3-319-30668-1_4.

[85] Bin Li et al. “Benign and Malignant Mammographic Image Classification Based on Convolutional Neural Networks”. In: *Proceedings of the 2018 10th International Conference on Machine Learning and Computing*. ICMLC 2018. Macau, China: Association for Computing Machinery, 2018, pp. 247–251. ISBN: 9781450363532. DOI: 10.1145/3195106.3195163.

[86] S. L. Li, L. Y. Zi, and B.H. Bin. “Facial Expression Recognition Based on Gabor Texture Features and Centre Binary Pattern”. In: *Applied Mechanics and Materials* 742 (2015), pp. 257–260.

[87] W. Li et al. “Automatic 4D Facial Expression Recognition Using Dynamic Geometrical Image Network”. In: *IEEE International Conference on Automatic Face Gesture Recognition*. May 2018, pp. 24–30. DOI: 10.1109/FG.2018.00014.

[88] S. X. Liao and M. Pawlak. “On image analysis by moments”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18.3 (1996), pp. 254–266.

[89] Y. Linde, A. Buzo, and R. Gray. “An Algorithm for Vector Quantizer Design”. In: *IEEE Transactions on Communications* 28.1 (Jan. 1980), pp. 84–95. ISSN: 0090-6778. DOI: 10.1109/TCOM.1980.1094577.

[90] Li Liu et al. “Extended local binary patterns for texture classification”. In: *Image and Vision Computing* 30 (Feb. 2012), pp. 86–99. DOI: 10.1016/j.imavis.2012.01.001.

[91] X. Lladó et al. “A textural approach for mass false positive reduction in mammography”. In: *Computerized Medical Imaging and Graphics* 33.6 (2009), pp. 415–422. ISSN: 0895-6111. DOI: https://doi.org/10.1016/j.compmedimag.2009.03.007.

[92] Ronald Lumia et al. “Texture analysis of aerial photographs”. In: *Pattern Recognition* 16.1 (Jan. 1983), pp. 39–46. DOI: 10.1016/0031-3203(83)90006-7.

[93] Y. Luo, T. Zhang, and Y. Zhang. “A novel fusion method of PCA and LDP for facial expression feature extraction”. In: *Optik* 127.2 (2016), pp. 718–721.

[94] B. Ma, W. Yuwei, and S. Fengyan. “Affine Object Tracking Using Kernel-Based Region Covariance Descriptors”. In: *Foundations of Intelligent Systems*. Vol. 122. 2012, pp. 613–623.

[95] D. Ma et al. “Facial expression recognition based on characteristics of block LGBP and sparse representation”. In: *Journal of Computational Methods in Sciences and Engineering* 15.3 (2015), pp. 537–547.

[96] J. Mairal, F. Bach, and J. Ponce. “Sparse modeling for image and vision processing”. In: *Foundations and Trends in Computer Graphics and Vision* 8.2-3 (2014), pp. 85–283.

[97] Ivy Majumdar, B.N. Chatterji, and Kar Avijit. “A Moment Based Feature Extraction for Texture Image Retrieval”. In: Jan. 2020, pp. 167–177. ISBN: 978-981-32-9452-3. DOI: 10.1007/978-981-32-9453-0_17.

[98] M. Mavadati, P. Sanger, and M. H. Mahoor. “Extended DISFA Dataset: Investigating Posed and Spontaneous Facial Expressions”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2016, pp. 1452–1459.

[99] S. M. Mavadati et al. “DISFA: A Spontaneous Facial Action Intensity Database”. In: *IEEE Transactions on Affective Computing* 4(2).2 (2013), pp. 151–160.

[100] Y. El Merabet, Y. Ruichek, and A. El Idrissi. “Attractive-and-repulsive center-symmetric local binary patterns for texture classification”. In: *Engineering Applications of Artificial Intelligence* 78 (2019), pp. 158–172. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2018.11.011. URL: http://www.sciencedirect.com/science/article/pii/S0952197618302495.

[101] Mohamed Meselhy Eltoukhy, Ibrahima Faye, and Brahim Belhaouari Samir. “A comparison of wavelet and curvelet for breast cancer diagnosis in digital mammogram”. In: *Computers in Biology and Medicine* 40.4 (2010), pp. 384–391. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2010.02.002.

[102] Mohamed Meselhy Eltoukhy, Ibrahima Faye, and Brahim Belhaouari Samir. “A statistical based feature extraction method for breast cancer diagnosis in digital mammogram using multiresolution representation”. In: *Computers in Biology and Medicine* 42.1 (2012), pp. 123–128. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2011.10.016.

[103] Figlu Mohanty, Suvendu Rup, and Bodhisattva Dash. “Automated diagnosis of breast cancer using parameter optimized kernel extreme learning machine”. In: *Biomedical Signal Processing and Control* 62 (2020), p. 102108. ISSN: 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2020.102108.

[104] Figlu Mohanty et al. “Digital mammogram classification using 2D-BDWT and GLCM features with FOA-based feature selection approach”. In: *Neural Computing and Applications* 32.11 (June 2020), pp. 7029–7043. ISSN: 1433-3058. DOI: 10.1007/s00521-019-04186-w.

[105] David J. Montana. “Strongly Typed Genetic Programming”. In: *Evolutionary Computation* 3.2 (June 1995), pp. 199–230. DOI: 10.1162/evco.1995.3.2.199.

[106] Shushi Namba et al. “Spontaneous Facial Expressions Are Different from Posed Facial Expressions: Morphological Properties and Dynamic Sequences”. In: *Current Psychology* 36 (May 2016). DOI: 10.1007/s12144-016-9448-9.

[107] Loris Nanni, Alessandra Lumini, and Sheryl Brahnam. “Survey on LBP based texture descriptors for image classification”. In: *Expert Systems with Applications* 39.3 (Feb. 2012), pp. 3634–3641. DOI: 10.1016/j.eswa.2011.09.054.

[108] Hieu V Nguyen and Li Bai. “Cosine similarity metric learning for face verification”. In: *Asian conference on computer vision*. Springer. 2010, pp. 709–720.

[109] R. A. Nurtanto Diaz, N. Nyoman Tria Swandewi, and K. D. Pradnyani Novianti. “Malignancy Determination Breast Cancer Based on Mammogram Image With K-Nearest Neighbor”. In: *2019 1st International Conference on Cybernetics and Intelligent System (ICORIS)*. Vol. 1. 2019, pp. 233–237. DOI: 10.1109/ICORIS.2019.8874873.

[110] T. Ojala, M. Pietikainen, and T. Maenpaa. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.7 (July 2002), pp. 971–987. DOI: 10.1109/tpami.2002.1017623.

[111] Timo Ojala, Matti Pietikäinen, and David Harwood. "A comparative study of texture measures with classification based on featured distributions". In: *Pattern Recognition* 29.1 (Jan. 1996), pp. 51–59. DOI: 10.1016/0031-3203(95)00067-4.

[112] Timo Ojala, Matti Pietikäinen, and T Maenpaa. "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24 (Aug. 2002), pp. 971–987. DOI: 10.1109/TPAMI.2002.1017623.

[113] K Palaniappan et al. "Efficient feature extraction and likelihood fusion for vehicle tracking in low frame rate airborne video". In: *2010 13th International Conference on Information Fusion*. IEEE, July 2010. DOI: 10.1109/icif.2010.5711891.

[114] Zhibin Pan, Xiuquan Wu, and Zhengyi Li. "Central pixel selection strategy based on local gray-value distribution by using gradient information to enhance LBP for texture classification". In: *Expert Systems with Applications* 120 (2019), pp. 319–334. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2018.11.041. URL: http://www.sciencedirect.com/science/article/pii/S0957417418307620.

[115] J. S. Payne, L. Heppelwhite, and T. J. Stonham. "Perceptually based metrics for the evaluation of textural image retrieval methods". In: *Proceedings IEEE International Conference on Multimedia Computing and Systems*. Vol. 2. June 1999, 793–797 vol.2. DOI: 10.1109/MMCS.1999.778587.

[116] P.J. Phillips and A. J. O. Toole. "A dynamic texture-based method to recognition of facial actions and their temporal models". In: *Pattern Analysis and Machine Intelligence, IEEE Transaction* (2010), pp. 1940–1954.

[117] P.J. Phillips and A.J.O. Toole. "Comparison of human and computer performance across face recognition experiments". In: *Image Vision Computing* 24 (2014), pp. 74–85.

[118] Richard Platania et al. "Automated Breast Cancer Diagnosis Using Deep Learning and Region of Interest Detection (BC-DROID)". In: *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology,and Health Informatics*. ACM-BCB '17. Boston, Massachusetts, USA: Association for Computing Machinery, 2017, pp. 536–543. ISBN: 9781450347228. DOI: 10.1145/3107411.3107484.

[119] S. J. Preece et al. "A Comparison of Feature Extraction Methods for the Classification of Dynamic Activities From Accelerometer Data". In: *IEEE Transactions on Biomedical Engineering* 56.3 (2009), pp. 871–879.

[120] L. R. Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition". In: *Proceedings of the IEEE* 77.2 (Feb. 1989), pp. 257–286. ISSN: 0018-9219. DOI: 10.1109/5.18626.

[121] Dina A. Ragab et al. “A framework for breast cancer classification using Multi-DCNNs”. In: *Computers in Biology and Medicine* 131 (2021), p. 104245. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2021.104245.

[122] Y. Rahulamathavan et al. “Facial Expression Recognition in the Encrypted Domain Based on Local Fisher Discriminant Analysis”. In: *IEEE Transactions on Affective Computing* 4(1).1 (2013), pp. 83–92.

[123] Georgia Rajamanoharan et al. “Recognition of 3D facial expression dynamics”. In: *Image and Vision Computing* 30 (Oct. 2012), pp. 762–773. DOI: 10.1016/j.imavis.2012.01.006.

[124] Vedantham Ramachandran and Edara Reddy. “A robust feature extraction with optimized DBN-SMO for facial expression recognition”. In: *Multimedia Tools and Applications* 79 (May 2020), pp. 21487–21512. DOI: https://doi.org/10.1007/s11042-020-08901-x.

[125] Essam Rashed and M. Samir Abou El Seoud. “Deep Learning Approach for Breast Cancer Diagnosis”. In: *Proceedings of the 2019 8th International Conference on Software and Information Engineering*. ICSIE ’19. Cairo, Egypt: Association for Computing Machinery, 2019, pp. 243–247. ISBN: 9781450361057. DOI: 10.1145/3328833.3328867.

[126] Essam A. Rashed, Ismail A. Ismail, and Sherif I. Zaki. “Multiresolution mammogram analysis in multilevel decomposition”. In: *Pattern Recognition Letters* 28.2 (2007), pp. 286–292. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2006.07.010.

[127] Irina Rish et al. “An empirical study of the naive Bayes classifier”. In: *IJCAI 2001 workshop on empirical methods in artificial intelligence*. Vol. 3. 22. 2001, pp. 41–46.

[128] A. Rosenfeld and M. Thurston. “Edge and Curve Detection for Visual Scene Analysis”. In: *IEEE Transactions on Computers* C-20.5 (1971), pp. 562–569.

[129] Anwar Saeed et al. “Frame-Based Facial Expression Recognition Using Geometrical Features”. In: *Advances in Human-Computer Interaction* 2014 (Apr. 2014), pp. 1–13. DOI: 10.1155/2014/408953.

[130] S.R. Safavian and D. Landgrebe. “A survey of decision tree classifier methodology”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 21.3 (1991), pp. 660–674. DOI: 10.1109/21.97458.

[131] Christos Sagonas et al. “300 Faces In-The-Wild Challenge: database and results”. In: *Image and Vision Computing* 47 (2016). 300-W, the First Automatic Facial Landmark Detection in-the-Wild Challenge, pp. 3–18. ISSN: 0262-8856. DOI: https://doi.org/10.1016/j.imavis.2016.01.002.

[132] Harith Al-Sahaf et al. “Extracting image features for classification by two-tier genetic programming”. In: *2012 IEEE Congress on Evolutionary Computation*. IEEE, June 2012. DOI: 10.1109/cec.2012.6256412.

[133] Albert Satorra and Peter M Bentler. “A scaled difference chi-square test statistic for moment structure analysis”. In: *Psychometrika* 66.4 (2001), pp. 507–514.

[134] Debashis Sen, Samyak Datta, and R. Balasubramanian. “Facial emotion classification using concatenated geometric and textural features”. In: *Multimedia Tools and Applications* 78.8 (Apr. 2019), pp. 10287–10323. ISSN: 1573-7721. DOI: 10.1007/s11042-018-6537-9. URL: https://doi.org/10.1007/s11042-018-6537-9.

[135] J. Shao et al. “3D dynamic facial expression recognition using low-resolution videos”. In: *Pattern Recognition Letters* 65 (2015), pp. 157–162.

[136] Punitha Stephan et al. “A hybrid artificial bee colony with whale optimization algorithm for improved breast cancer diagnosis”. In: *Neural Computing and Applications* (May 2021).

[137] Y. Sun and L. Yin. “Facial expression recognition based on 3D dynamic range model sequences”. In: *ECCV 2008. Lecture Notes in Computer Science.* (2008), pp. 58–71.

[138] Y. Sun et al. “Tracking Vertex Flow and Model Adaptation for Three-Dimensional Spatiotemporal Face Analysis”. In: *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 40.3 (2010), pp. 461–474.

[139] J.A.K. Suykens and J. Vandewalle. In: *Neural Processing Letters* 9.3 (1999), pp. 293–300. DOI: 10.1023/a:1018628609742.

[140] Roy Swalpa Kumar et al. “Local morphological pattern: A scale space shape descriptor for texture classification”. In: *Digital Signal Processing* 82 (2018), pp. 152–165. ISSN: 1051-2004. DOI: https://doi.org/10.1016/j.dsp.2018.06.016.

[141] Roy Swalpa Kumar et al. “Unconstrained texture classification using efficient jet texton learning”. In: *Applied Soft Computing* 86 (2020), p. 105910. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2019.105910.

[142] H. Tang and T. S. Huang. “3D facial expression recognition based on automatically selected features”. In: *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. 2008, pp. 1–8.

[143] Øivind Due Trier, Anil K Jain, and Torfinn Taxt. “Feature extraction methods for character recognition-a survey”. In: *Pattern recognition* 29.4 (1996), pp. 641–662.

[144] Tuncer Turker, Dogan Sengul, and Ertam Fatih. “A novel neural network based image descriptor for texture classification”. In: *Physica A: Statistical Mechanics and its Applications* 526 (2019), p. 120955. ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2019.04.191.

[145] O. Tuzel, F. Porikli, and P. Meer. “Region Covariance: A Fast Descriptor for Detection and Classification”. In: *Computer Vision – ECCV 2006*. 2006, pp. 589–600.

[146] 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*. 2005, pp. 76–76.

[147] R. Vijayarajeswari et al. “Classification of mammogram for early detection of breast cancer using SVM classifier and Hough transform”. In: *Measurement* 146 (2019), pp. 800–805. ISSN: 0263-2241. DOI: https://doi.org/10.1016/j.measurement.2019.05.083.

[148] S. Wang et al. “A natural visible and infrared facial expression database for expression recognition and emotion inference”. In: *IEEE Transactions on Multimedia* 12.7 (2010), pp. 682–691.

[149] Yi Wang, Wenke Yu, and Zhice Fang. “Multiple Kernel-Based SVM Classification of Hyperspectral Images by Combining Spectral, Spatial, and Semantic Information”. In: *Remote Sensing* 12.1 (2020). ISSN: 2072-4292. DOI: 10.3390/rs12010120.

[150] X. Wei et al. “Unsupervised Domain Adaptation with Regularized Optimal Transport for Multimodal 2D+3D Facial Expression Recognition”. In: *IEEE International Conference on Automatic Face Gesture Recognition*. May 2018, pp. 31–37. DOI: 10.1109/FG.2018.00015.

[151] N. Werghi et al. “Boosting 3D LBP-based Face Recognition by Fusing Shape and Texture Descriptors on the Mesh”. In: *IEEE Transaction on Information Forensics and Security* 11 (2016), pp. 964–979.

[152] Xiaosheng Wu and Junding Sun. “Joint-scale LBP: a new feature descriptor for texture classification”. In: *The visual computer* 33.3 (2017), pp. 317–329. DOI: 10.1007/s00371-015-1202-z.

[153] H. Xiaopeng et al. “Sigma Set: A small second order statistical region descriptor”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. June 2009, pp. 1802–1809.

[154] Siyue Xie, Haifeng Hu, and Yongbo Wu. “Deep multi-path convolutional neural network joint with salient region attention for facial expression recognition”. In: *Pattern Recognition* 92 (2019), pp. 177–191. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2019.03.019.

[155] M. Xue et al. “Automatic 4D Facial Expression Recognition Using DCT Features”. In: *IEEE Winter Conference Applications of Computer Vision* (2015), pp. 199–206.

[156] Huiyuan Yang, Umur Ciftci, and Lijun Yin. “Facial Expression Recognition by De-expression Residue Learning”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018, pp. 2168–2177. DOI: 10.1109/CVPR.2018.00231.

[157] Jiajia Yang and Shangfei Wang. “Capturing Spatial and Temporal Patterns for Distinguishing between Posed and Spontaneous Expressions”. In: *Proceedings of the 25th ACM international conference on Multimedia*. Oct. 2017, pp. 469–477. DOI: 10.1145/3123266.3123350.

[158] L. Yin et al. “A high-resolution 3D dynamic facial expression database”. In: *IEEE Automatic Face and Gesture Recognition* (2008).

[159] J. Yu, K. Ko, and K. Sim. “Facial point classifier using convolution neural network and cascade facial point detector”. In: *Journal of Institute of Control, Robotics and Systems* 22.3 (2016), pp. 241–246.

[160] S. Zafeiriou and I. Pitas. “Discriminant Graph Structures for Facial Expression Recognition”. In: *IEEE Transactions on Multimedia* 10(8).8 (2008), pp. 1528–1540.

[161] Feifei Zhang et al. “Joint Pose and Expression Modeling for Facial Expression Recognition”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018, pp. 3359–3368. DOI: 10.1109/CVPR.2018.00354.

[162] Hepeng Zhang, Bin Huang, and Guohui Tian. “Facial expression recognition based on deep convolution long short-term memory networks of double-channel weighted mixture”. In: *Pattern Recognition Letters* 131 (2020), pp. 128–134. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2019.12.013.

[163] Xinwei Zhang et al. “Pattern understanding and synthesis based on layout tree descriptor”. In: *The Visual Computer* (2019), pp. 1–15.

[164] S. Zhao, H. Yao, and X. Sun. “Video classification and recommendation based on affective analysis of viewers”. In: *Neurocomputing* 119 (2013), pp. 101–110.

[165] S. Zhao et al. “Approximating discrete probability distribution of image emotions by multi-modal features fusion”. In: *International Joint Conference on Artificial Intelligence*. 2017, pp. 4669–4675.

[166] Q. Zhen et al. “Muscular Movement Model-Based Automatic 3D/4D Facial Expression Recognition”. In: *IEEE Transactions on Multimedia* 18.7 (July 2016), pp. 1438–1450.

[167] Q. Zhen et al. “Magnifying Subtle Facial Motions for Effective 4D Expression Recognition”. In: *IEEE Transactions on Affective Computing* (2017), pp. 2252–2257.