Ministry of Higher Education and
Scientific Research
University of Babylon
College of Science for Women
Department of Computer Science

Abnormal Gait Patterns Classification Based on Spatial Image Features and Deep Learning Technique

A Thesis

Submitted to the Council of the College of Science for Women at University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master in Science \ Computer Science

By
Zainab Ali Abd Alhuseen Qamar

Supervised by

Prof.Dr. Mohammed Abdullah Naser & Lect. Dr. Fanar Ali Joda

2024 A.D. 1445 A.H.

بِسْمِ اللهِ الرَّحْمنِ الرَّحِيمِ

((قَالُوا سُبْحَانَكَ لَا عِلْمَ لَنَا إِلَّا مَا عَلَّمْتَنَا إِنَّكَ أَنْتَ الْعَلِيمُ

الْحَكِيمُ))

صدق الله العلي العظيم

سورة البقرة/الاية 32

Supervisors Certification

We certify that this thesis entitled **"Abnormal Gait Patterns Classification Based on Spatial Image Features and Deep Learning Technique"** is done by **(Zainab Ali Abd Alhuseen Qamar)** under our supervision.

Signature:
Name: prof. Dr. Mohammed Abdullah Naser
Date: / / 2024
Address: University of Babylon/College of Science for Women

Signature:
Name: Lect. Dr. Fanar Ali Joda
Date: / / 2024
Address: Directorate of Education of Babylon

Head of the Department Certification

In view of the available recommendations, I forward the thesis entitled **"Abnormal Gait Patterns Classification Based on Spatial Image Features and Deep Learning Technique"** for debate by the examining committee.

Signature:

Name: Asst. prof. Dr. Saif Mahmood Khalaf

Date: / / 2024

Address: University of Babylon/College of Science for Women

Dedication

To the soul of who wished to see me in this place and receive my degree

.. My father, may God have mercy on him

To the one who gave me safety and tenderness and brought me to this bright path in my life that carried me through illness and health and during my career

.. My dear mother

To my companion and my support in my life, and a mixture of father, brother and friend

.. My dear husband

To a piece of my heart and the generation of tomorrow

.. My children Ali and Hussein

To everyone who has been so supportive and encouraging throughout this entire journey

.. My sisters, brothers and friends

And to all of them I dedicate this humble work

Zainab

2024

Acknowledgements

Thanks to our God for giving me health and desire to complete this work and for everything.

Especial gratefulness and deep gratitude is due to my supervisors **Dr. Mohammed A. Naser and Dr. Fanar Ali Joda** for their encouragement, guidance and support during my research work. I wish to thank **all the staff members in the Computer Science Department**.

I also would like to express my wholehearted thanks to my family (**father, mother, My Husband, My children, sisters, and brothers**) for their patience and support. Their encouragement has been my greatest strength which enabled me to complete my research.

I would like to thank my colleagues for, who gave me their help or advice.

Zainab

2024

Abstract

Researchers have recently shown widespread interest in classifying abnormal behaviors using computer vision, in order to enhance security, safety, and efficiency in various fields. Researchers continue to explore and improve technologies to make these systems more effective and reliable.

The proposed system comes as an attempt to classify abnormal gait patterns based on gait analysis and walking scenes, where people's gaits are analyzed in several situations. Therefore, it works to provide a model for detecting deviations in individuals' behavior based on walking patterns.

This system works on a set of videos of data collected for people suffering from certain diseases that affect their walking and other healthy people, and performs a number of pre-processes on it, then performs a comprehensive analysis of a number of feature extraction techniques to classify images, with the aim of comparing and evaluating the performance of these techniques, which are: Histogram of Oriented Gradients (HOG), Local Binary Pattern (LBP), and Center-Symmetric Local Binary Pattern (CS-LBP), and Extended Center-Symmetric Local Binary Pattern (XCS-LBP). After obtaining the extracted features, the last and most important stage in the system is entered, which is the classification stage using one of the deep learning algorithms, which is (CNN+Vgg16).

To assess the efficiency of the introduced model and perform the evaluation on a variety of collected data sets including instances of normal and abnormal behavior. The system achieved a well performance with an accuracy of up to 99% using the CS-LBP and XCS-LBP methods. Experimental results indicate that these two methods are well at accelerating training and testing times, and experiments also showed that gait feature extraction methods played a role in the success of the system and in classify abnormal gait patterns.

List of Contents

Chapter three

Proposed System

Chapter four

Experiments Results and Discussions

Chapter five

Conclusions and Suggestions for Future Work

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

Abbreviation	Description
ACC	Accuracy
Adam	Adaptive Moment Estimation
AI	Artificial Intelligence
CNN	Convolutional Neural Network
CS-LBP	Center-Symmetric Local Binary Pattern
DL	Deep Learning
FC	Fully Connected
FE	Feature Extraction
FN	False Negative
FP	False Positive
HGA	Human Gait Analysis
HOG	Histogram of Oriented Gradients
LBP	Local Binary Pattern
LSFE	Local Spatial Feature Extraction
ML	Machine Learning
MSE	Mean Square Error
Relu	Rectified Linear Unit
RGB	Red Green Blue
TN	True Negative
TP	True Positive
VGG16	Visual Geometry Group 16
XCS-LBP	Extended Center-Symmetric Local Binary Pattern

Chapter One

General Introduction

Chapter One
General Introduction

1.1 Introduction

Human gait is a biometric property that can be used to determine the behavior of people based on their gait pattern, but it needs a robust security system in the current technological context. Human gait analysis is an emerging discipline in computer vision with important applications in areas such as osteoporosis diagnosis and patient monitoring[1], and also began to learn to walk it is used in practical cases in criminal investigations [2].

Nowadays, visual monitoring of public security is becoming increasingly important. As video surveillance data grows more and more, it is necessary to develop an automated visual monitoring system that can classify abnormal behaviors. Anyone who acts contrary to normal walking patterns is detected as abnormal behavior. Abnormal behavior is defined as “not normal, typical, stereotyped, or deviating from the normal norm.” The psychologist defines abnormal actions as deviating from accepted social norms of behavior [3].

With the development and expansion of the area of computer vision, a vast number of novel monitoring technologies have developed and attracted the attention of academics due to their broad range of monitoring applications [4][5]. Computer vision is a fast-increasing topic, and many new monitoring technologies have caught the interest of academics due to their applicability in a variety of sectors. For the sake of maintaining public safety, these cameras have been placed in different areas. This is why we want to create strategies for using these cameras to detect any strange behavior. An intelligent monitoring system can monitor the patient and identify numerous disorders connected with these people's gait activities, which may be

utilized to avoid such deadly incidents; Or, in terms of health, violence is aberrant behavior and activity, including the use of physical force to cause property damage or to kill or injure a person or animal [6].

The main objective of video surveillance is to detect, classify and recognize the interesting patterns [7]. Human viewing surveillance films to classify abnormal behavior are too slow, causing damage and injury[8]. There is a need for an automated system to classify abnormal behavior based on human gait and using deep learning methods.

1.2 Literatures Review

Many researches and development have gone into using computer vision to spot anomalies in human gait. A few of articles that have been written on this topic are summarized here.

In a 2016 study by Kohei Shiraga et al., they proposed a method for gait recognition using a Convolutional Neural Network (CNN). Where they feed files the most widespread image-based gait representation, Gait energy image (GEI) as input to the CNN designed for Gait recognition is called GEINet. More specifically, GEINet consists of two consecutive triplets of convolution, pooling, and normalizing layers, and two fully consecutive layers connected classes, which produce a set of similarities in individual training topics. They experimented with demonstrate the effectiveness of the proposed method recognizable terms for walking across the width in both the co-op and the non-cooperative settings that use a large population OU-ISIR dataset [9].

A deep convolutional neural network, built with fully connected layers and convolutional pooling, was used in the system proposed by Nithyakani P. et al. (2019). By training the neural network architecture using a gait energy image, they extracted information about a person's gait using normalization layers that accept

GEI data as input. This research aimed to secretly identify individuals by analyzing their walking patterns. The dataset was TUM GAID [10].

Gait analysis using sensor technologies, including a wearable inertial measurement unit (IMU), was presented in a paper by Jing Gao et al. (2019). Hemiplegic walking, tiptoeing, and threshold walking are just a few of the identifiable abnormal gait patterns. First, to identify and name the abnormal gait, they proposed using long-term memory and convolutional neural network (LCWSnet) technology. They enhanced the convergence layer of the LSTM-CNN model and the model's ability to classify abnormal gait. Experimental results show that their proposed approach based on LCWSnet is able to identify gait abnormalities in the data [11].

Ahmed R. Hawass et al. (2019), their study aimed to identify individuals based on their style and gait. They efficiently used deep learning models in gait recognition systems. Gait features were automatically measured while walking for the recognition system. Two vital modules were used, motion detection and tracking and feature extraction. Accordingly, the main unit distinguishes the walking pattern in the photo or video sequence. The second module, "Feature Extraction," extracts features from the silhouette sequence. Finally, the CNN is fed in to build unique features. These features are used to train the neural network, and the evaluation was performed on common geometry datasets [12].

A. Mehmood (2021), in this study, he presented the CNN-based method that effectively determines and categorizes if a video clip contains anomalous human behaviors including falling, loitering, and violence in open spaces. In order to improve prediction performance, this method employs a two-stream architecture employing two separate 3D CNNs that receive both optical and video stream input.

The model was trained using a customized data set suited for each anomalous behavior after transfer learning was applied [13].

Faizan Saleem et al. (2021), this study stated the use of deep learning and feature optimization, and a new framework of human gait recognition was provided. Feature extraction, feature selection, feature merging, and classification are all parts of their proposed framework. They used three different heart procedures during the improvement phase. Inception-ResNet-V2 and NASNet Mobile, two pre-trained models, were used to achieve this. The two models were trained and fine-tuned using the CASIA B gait dataset. The best deep models were chosen after several deep models properties were improved utilizing a modified three-step whale optimization process. Use the MDESF (modified mean absolute deviation for extended serial fusion) approach to include the most desired features. The final categorization was then determined utilizing a variety of unique approaches [14].

Khan, Muhammed A, et al. (2021), this study the purpose of improving the performance of a Human Gait Analysis (HGA) system, they proposed a framework that relies heavily on deep feature clustering and is powered by Kernel Extreme Machine Learning (KELM). To select the most useful and efficient deep features, they proposed a new process they called Euclidean criteria, which involves maximizing the geometric mean and conditional entropy. To group similar features together, they used canonical correction analysis (CCA). The proposed method was tested using the CASIA B gait database, which is open to the public [15].

P. Kuppusamy and V.C. Bharathi (2022), in this study investigates abnormal human behaviors using various CNNs to identify abnormal behaviors in the video. This study noted that the three-dimensional convolutional neural network better performance of machine learning algorithms. The comparison showed that different

CNN models each shaped to identify different abnormal human behaviors with diverse datasets [16].

The authors C. Filipi Gonçalves dos Santos et al. (2022) in this study, they synthesize and analyze existing research on biometric detection using gait recognition, focusing on deep learning algorithms, highlighting their advantages, and addressing their drawbacks. In addition, they provided categorized and described descriptions of the datasets, methods, and architectures used to address the relevant limitations [17].

In 2023, P. Chophuk et al. presented gait patterns utilizing motion-based AI from a fixed camera in their investigation of a theft detection system. This technique uses the Yolov5 model for object localization, which has less of an impact on the issue of recognizing moving objects from RGB video in the background. The box centroid is then determined using the bounding boxes of Yolov 5 and utilized to create a walking pattern behavior movement. Finally, the sequences of the motion patterns utilized in tandem with a two-way long-term memory network. The datasets include 30 films of people walking normally and 30 movies of people stealing while walking [18].

Table (1.1): The Summary of Related Works

Reference	Research Title	Technique Used	Dataset	Accuracy
P. Nithyakani et al., in 2019 [10]	Human Gait Recognition using Deep Convolutional Neural Network	Deep CNN	TUM Gaid	94.7 %

Reference	Research Title	Technique Used	Dataset	Accuracy
J. Gao et al., in 2019 [11]	Abnormal Gait Recognition Algorithm Based on LSTM-CNN Fusion Network	LSTM-CNN	manually collected from the inertia sensors	93.1 %
A. R. Hawas et al., in 2019 [12]	Gait identification by convolutional neural networks and optical flow	CNN	CASIA B Gait	95 %
F. Saleem et al., in 2021 [14]	Human gait recognition: A single stream optimal deep learning features fusion	CNN	CASIA B Gait	89 %
M. A. Khan et al., in 2021 [15]	Human gait analysis for osteoarthritis prediction: a framework of deep learning and kernel extreme learning machine	CNN	CASIA B Gait	96 %
P. Chophuk et al. in 2023 [18]	Theft Detection by Patterns of Walking Behavior using Motion-based Artificial Intelligence	Bi-LSTM	UCF-Crime	97 %

1.3 Problem Statement

There are many problems facing the task of classifying abnormal behaviors that depend on the way of walking, and the most important of these challenges are:

1. Each person has his own biological characteristics, and deviation from normal

walking patterns can indicate abnormal behavior. The human elements and traditional methods that review surveillance films are very slow, causing damage and injuries on the health side and the delay in discovering the patient's behavior in its early stages from The way he walks leads to health damage, aggravation of the condition, and his failure to receive treatment in a timely manner.

2. The important problems in such research are how to select and extract the important features that lead us to obtain the correct classification results. System accuracy and calculation time are impacted by irrelevant feature data collected from the original frames.
3. It may be difficult to obtain sufficient and diverse data for individuals with walking problems. Variation in walking positions and surrounding conditions can be challenging to understand the full context of movement.

   Therefore, it is very important to suggest the appropriate technique to overcome the above the challenge.

1.4 The Aim & Objectives of Thesis

The objectives of this thesis can be summarized as follows:

1. Developing a model for detecting abnormal behavior in individuals based on their walking patterns (gait) using a dataset of individuals with diverse behaviors. The objective is to leverage the features of walking to accurately classify human behavior as normal or abnormal.
2. Identifying and implementing effective algorithms for extracting important characteristics from images of walking patterns, while eliminating irrelevant or unimportant characteristics. The objective is to enhance the accuracy of the classification stage by focusing on relevant features.
3. Utilizeing a convolutional neural network (CNN+Vgg16) as the primary

classification approach, aiming to reduce execution time and improve classification accuracy. The objective is to exploit the effectiveness of CNNs in classification tasks, thereby achieving efficient and accurate classification of abnormal behavior.

1.5 Research Impact

The research conducted in this thesis has the potential to make a significant impact in various fields, especially in the health field, as early detection of pathological conditions through gait analysis can lead to timely diagnosis and appropriate treatment. By providing a reliable model for classifying abnormal behavior, healthcare professionals can identify potential health problems at an early stage, allowing for more effective intervention and management. This can improve patient outcomes, reduce health care costs, and enhance overall health.

1.6 Thesis Layout

The thesis has five chapters and briefly gives their description below. Next, to this chapter, the remaining parts of this thesis are distributed among the following chapters:

- **Chapter Two: (**Theoretical Background**)**

This chapter presents an introduction, abnormal behavior and the relationship of human gait in detecting this behavior, and explains all the techniques and methods used in the proposed system.

- **Chapter Three:** (The Proposed System)

This chapter presents a proposed system for detecting abnormal behavior based on human gait using deep learning techniques and explains the stages of the designed system.

- **Chapter Four:** (Experiments Results and Discussion)

This chapter presents the description of the different experiments and discusses the results and evaluations obtained from the implementation of the proposed system.

- **Chapter Five:** (Conclusions and Suggestions for Future Works)

This chapter presents the main conclusions and suggestions for future action from the results of the proposed system.

Chapter Two

Theoretical Background

Chapter Two

Theoretical Background

2.1 Introduction

The classification of abnormal gait of pedestrians has gradually become a popular topic in the field of intelligent video surveillance [19]. The amount of surveillance data and internet data is increasing more rapidly in recent years[20]. Previous safety management schemes on artificial operation can not satisfy the era of big data [21]. Nowadays, intelligence is in great demand efficient and reliable video surveillance system, And intelligent video surveillance system needs to be improved [22]. The biometric feature was used, which is the human gait through his gait, and using deep learning, the person's behavior is classified as normal or abnormal. The data entry must have clear extracted features in order for the classification process to be good and with high accuracy.

In this chapter, the theoretical background of this thesis framework is presented which includes a review of biometric systems and in particular the concept of gait analysis and abnormal gait, an explanation of a set of methodology and pre-processing methods for extracting features that can be applied to images, and a brief description of the concept and methodology of deep learning and its models. Furthermore, a set of performance appraisal criteria is spelled out. These terms are the main outlines that will be presented and discussed and that have an important role in achieving the motivation behind the message.

2.2 Gait Abnormal patterns

“Behavior that is statistically atypical or unfamiliar within a given culture, maladaptive, or harmful to an individual or those around him” is a definition of

abnormal behaviour. There are two classes of irregular behavior: global irregular actions and local irregular actions. Local abnormal behavior includes one or two individuals who behave differently from the rest of the population (eg, loitering, falling over, fighting, sick, etc.). A deviant global action includes many individuals or organizations whose actions may be abnormal, such as acts of violence, actions taken to flee a natural disaster, conflict, and so on. Action representation and abnormal action evaluation are two of the modules that make up abnormal behavior recognition [16].

Walking and running steadily are seen as normal activities, and walking upright is one such activity signs of normal human appearance. But there is also other behaviours. For example, when an event occurs like a stomach ache occurs, the person stoops and he walks slowly or he may bend and suddenly, like squatting while walking, so we define this behavior as abnormal, or the movement of the limbs of a person with a certain disease such as Parkinson's disease, this person's gait is classified as abnormal [23].

However, there is no single accepted definition of abnormal behavior. Some academics assert that any behavior that deviates from standard procedures is abnormal. According to some academics, deviant behaviors are those that occur rarely or only for a short time. In Figure (2.1) some cases showing normal and abnormal gait behavior.

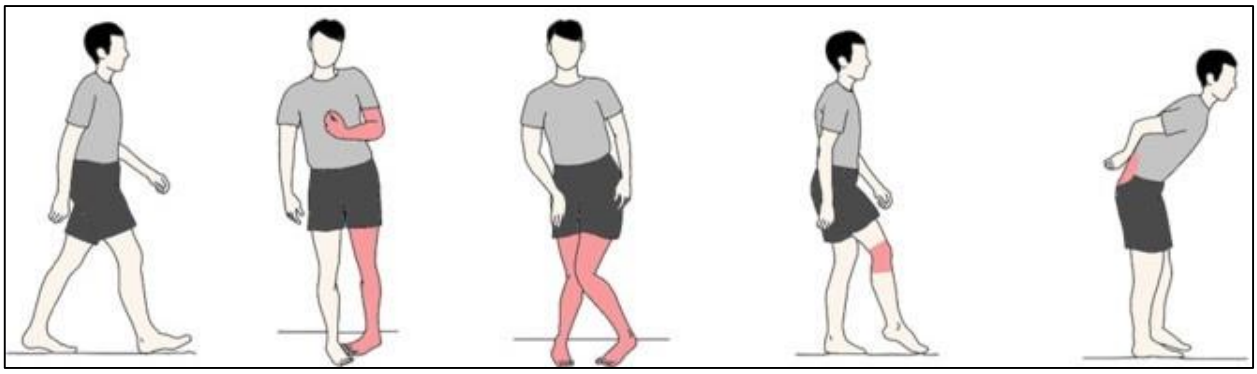


Figure (2.1): Normal And Abnormal Gait Patterns [24].

2.3 Biometric Systems

Biometric systems have gained popularity in the field of information technology due to their ability to provide secure and reliable identification and authentication. Biometrics refers to the measurement and analysis of an individual's unique physical or behavioral characteristics.

The term "biometric" is derived from the Greek words "bio" meaning life, and "metric" meaning to measure [25]. Biometric systems utilize an individual's unique physical or behavioral characteristics for identification and authentication purposes.

Some commonly used biometric characteristics include [26]:

1. **Finger Print:** are unique patterns of ridges and valleys on the skin of the fingers and palms of humans. They are used for identification purposes, particularly in forensic science and law enforcement.
2. **Face:** Facial recognition technology analyzes the unique features of a person's face, such as the distance between the eyes, shape of the nose, and contour of the jawline, to verify their identity.
3. **Iris:** Iris recognition technology uses the patterns in the colored portion of the eye, known as the iris, to authenticate individuals. The iris has unique characteristics that can be used for accurate identification.

4. **Retina:** Retinal scanning involves capturing the unique patterns of blood vessels in the back of the eye. This technique provides a highly secure method of identification.
5. **Palm Print:** Palm print recognition analyzes the patterns present on the palm of a person's hand, including ridges, lines, and creases. These patterns are unique and can be used for identification.
6. **Speech Pattern:** Voice recognition technology analyzes the unique characteristics of an individual's speech, including pitch, tone, and pronunciation, to verify their identity.
7. **Signature:** Signature recognition involves capturing and analyzing an individual's handwritten signature. The dynamics of the signature, such as speed, pressure, and stroke order, can be used for authentication.
8. **Gait:** Gait recognition technology analyzes an individual's walking pattern and movement characteristics. The way a person walks is unique and can be used for identification and detection of abnormal behavior.

The use of biometric characteristics in a system provides several advantages over traditional identification methods such as PINs and passwords. Biometric traits cannot be easily stolen, lost, or replicated by an unauthorized individual [27]. This enhances the security of the system and reduces the risk of fraud. Additionally, biometric characteristics are unique to each individual, making it difficult for an impostor to gain unauthorized access.

It should be noted that although biometric systems provide enhanced security, they also raise privacy and data protection concerns. The storage and processing of vital data requires strict security measures to ensure that the information is adequately protected, Figure (2.2) represents some type commonly used biometrics.

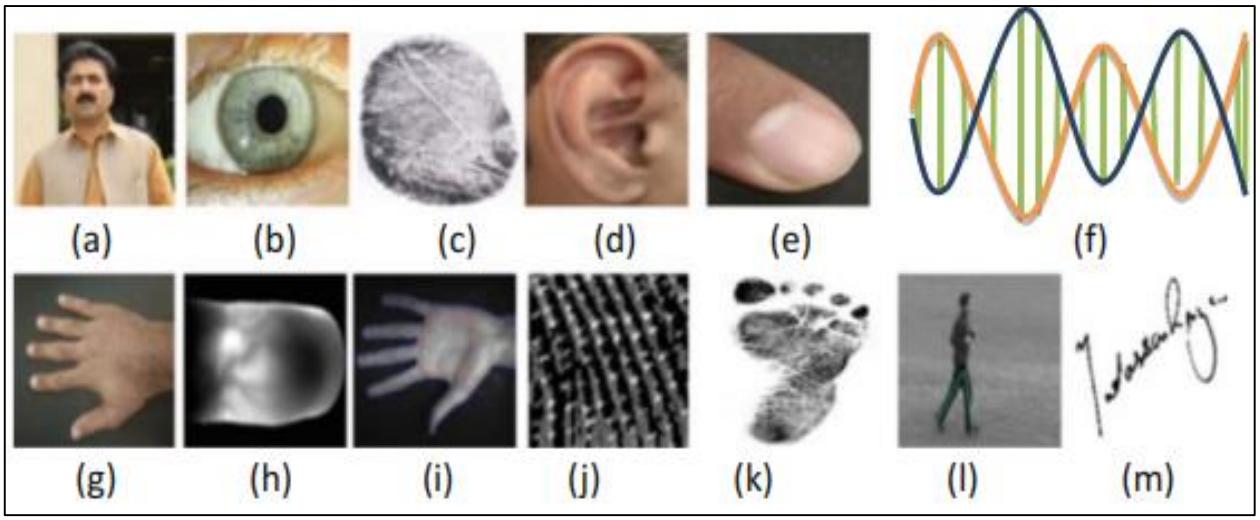


Figure (2.2): Biometric Types (a) face (b) iris (c) finger print (d) ear (e) nail bit (f) DNA (g) hand (h) finger vein (i) Palm vein (j) sweat pores (k) foot print (l) gait (m) signature [28]**.**

2.4 Gait Analysis

Gait analysis is an assessment of the way the body moves, usually by walking or running, from one place to another. The purpose of gait analysis is to detect any abnormalities in locomotion. Gait analysis refers to the process of recognizing individuals based on their unique walking style [29]. It is a biometric technology that uses contactless data acquisition, which means that it does not require physical contact with the person being identified.

Physiological studies have shown that each individual has a distinctive gait style that is difficult to mimic or mimic accurately. This uniqueness makes walking a valuable biometric for applications such as video surveillance in security systems and locating missing individuals. Video sequences captured from multiple cameras and different angles can be used to identify a human based on how they walk, even in crowded or densely populated environments [30].

To achieve accurate gait recognition, researchers and developers need to consider these factors and develop algorithms and systems that can effectively deal with

factors that can affect gait recognition. These factors include various carrying conditions (such as carrying bags or other objects), the clothes people wear, And the effects of lighting that can affect the vision of a person's gait and the speed of the gamete [31]. This ongoing research aims to improve the reliability and applicability of gait classification technology in real-world scenarios.

2.5 Feature Extraction Techniques

Feature extraction plays a vital role in various applications such as diagnosis, classification, clustering, recognition, and detection [32]. It involves selecting relevant information from raw data and transforming it into a more compact and meaningful representation, known as features. These features capture the essential characteristics of the data that are relevant to the specific task at hand.

The choice of suitable features is crucial because it directly affects the performance and accuracy of the application. Well-designed features can enhance the effectiveness of machine learning algorithms and enable them to learn more efficiently and effectively.

It's worth mentioning that feature extraction is an iterative process, where researchers often refine and improve the feature set to achieve better performance. Feature engineering, which involves designing and selecting relevant features, requires domain expertise and a deep understanding of the data and the problem at hand.

By selecting suitable features, researchers can improve the accuracy and efficiency of various applications, enabling better diagnosis, classification, clustering, recognition, and detection tasks. The choice of feature extraction technique depends on factors such as the nature of the data, available computational resources, and the specific requirements of the application. Researchers typically experiment with

different feature extraction methods and evaluate their impact on the performance of the overall system.

A common representation of feature extraction methods ara frequency domain and spatial domain which are representations of signals or images commonly used in signal processing and image processing. Each domain has its own set of features that provide different insights into the properties of the data [33].

2.5.1 Frequency Domain Features

Frequency domain features are mathematical representations of signals or data in the frequency domain. The frequency domain is a way of analyzing and characterizing signals based on their frequency content. It provides insights into the various frequency components present in a signal and their respective magnitudes [34]. Which is the idea to transform images into the frequency domain to detect artifacts is well known. It is used since the mid-1990s for detection of artifacts created by watermarking embedding techniques as described and also for a morph detection approach in [35].

2.5.2 Spatial Domain Features

In image or spatial data analysis, features like texture, color histograms, edge information, or shape descriptors can be extracted to represent the visual properties of the data. In image processing and computer vision, the spatial domain refers to the representation and manipulation of an image in its original pixel coordinates. It deals with the raw pixel values and their spatial arrangement within an image. In other words, the spatial domain focuses on the individual pixels of an image and their positions relative to each other.

When an image is represented in the spatial domain, each pixel is typically assigned a value that corresponds to its color or intensity. By analyzing and

manipulating these pixel values, various operations can be performed, such as filtering, enhancement, segmentation, and object recognition. Spatial domain operations directly manipulate the pixel values themselves, often by applying mathematical operations or filters to individual or groups of pixels. Examples of spatial domain operations include blurring, sharpening, edge detection, noise reduction, and contrast enhancement.

The use of local spatial feature extraction (LSFE) techniques. It will be easier to apply local descriptors to the image and give it more meaning once it is optimized and converted to a grayscale image. The (HOG) method was used to extract the features at the pixel level. and extract the features again at the block level using different methods (LBP, CS-LBP and XCS-LBP), which are discussed below:

2.5.2.1 Extract Features at the Pixel Level

Features extracted at the pixel level in which the output value at a given coordinate depends only on the input value at the same coordinate.

- **Histogram of Oriented Gradients (HOG)**

It is a property descriptor that was initially proposed for pedestrian detection [36]. Counts the number of times the gradient direction occurs in the detection window. It works by dividing the image into small windows and calculating graphs of the direction of the gradient within each window. The histograms are then connected to form a feature vector that can be used to classify the images. The following summarizes the primary stage in computing HOG features [37]:

- ***Gradient calculation***: In this stage, spatial gradients are calculated in the horizontal and vertical directions. These two gradations are then used to calculate the magnitudes and angles of the gradient, as in equations (2.1), (2.2):

$$Gx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad (2.1)$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad (2.2)$$

Where Gx and Gy are the corresponding gradients in the horizontal and vertical directions. Notably, pictures may be convolved using the Sobel mask to achieve equations (2.1), (2.2) computations. It computes the gradient of picture intensity at each pixel in the image to determine how it functions. To find edges in pictures, it is frequently used in image analysis. This technique may be influenced by the center difference, which, on average, favors the center pixels. It combines distinction and smoothing. The gradient of the picture is determined for each pixel point in the image for Sobel edge detection. Next, we may use equations (2.3), (2.4) to determine the gradient's size and direction:

$$M = |G_x| + |G_y| \qquad (2.3)$$

$$\theta(x, y) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \qquad (2.4)$$

Where m = magnitude and θ(x,y) = the direction of the gradient.

- ***Orientation binning***: The picture is separated into tiny, interconnected areas called cells during this stage. According to the gradient angle, the gradient magnitude of every pixel in a cell is divided into several orientation bins.

- This stage involves grouping contiguous cells into blocks and normalizing each block. In a detection window, a descriptor is created by concatenating the normalized block histograms.

2.5.2.2 Extract Features at the Block Level

Features extracted at the block level in which the output value at a given coordinate depends on the input values in the vicinity of the same coordinate.

- **Local Binary Pattern (LBP)**

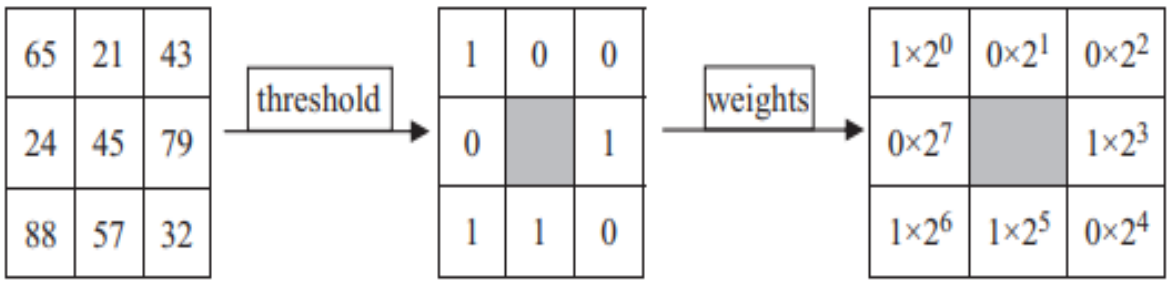
It was first introduced by Ojala et al [38]. It is a potent technique for describing textures that is based on statistical analysis and demonstrates how useful it is for this purpose. Despite the fact that several LBP variations are often utilized for face analysis because to their effective classification performance, they have not yet been demonstrated to be compact. Therefore, feature fusion is thought to be a successful strategy. LBP that maximizes mutual information performs better when analyzing faces [39]. By comparing the gray level intensity of the center pixel to its surrounding neighbors, it is calculated. Each neighbor is given a binary value of 1 or 0, depending on whether it is brighter or darker than the central pixel. After that, the binary values are combined to create a binary pattern that may be utilized as an image feature. The LBP result might be expressed as follows using equations (2.5), (2.6) [40]:

$$LBP_{P,R}(x_c, y_c) = \sum_{n=0}^{7} \delta\,(g_n - g_c)^{2^n} \qquad (2.5)$$

Where g_c = center pixel value positioned at (x_c, y_c), g_n = one of the eight surrounding center pixel values with the radius R, P = is the whole neighbourhood number, and a sign function δ(.) is defined such that

$$\delta(x)\begin{cases} 1 & if\ x \geq 0 \\ 0 & otherwise. \end{cases} \qquad (2.6)$$

Figure (2.3) shows a simple example of applying the LBP method as follows:



LBP code of the center pixel **45**:$1\times2^0+0\times2^1+0\times2^2+1\times2^3+0\times2^4+1\times2^5+1\times2^6+0\times2^7=1+8+32+64=$**105**

Figure (2.3): An example of the LBP [41]

- **Center-Symmetric Local Binary Pattern (CS-LBP)**

It represents yet another modified version of LBP. It is originally proposed to alleviate some drawbacks of the standard LBP [42]. In CS-LBP, the center symmetric pairs of pixels are compared rather than the gray levels of each pixel and the center pixel. It has a tight connection to the gradient operator. It takes into account the variations in gray scale between adjacent pairs of pixels in a neighborhood. Therefore, CS-LBP utilizes both gradient-based features and LBP. The edges and prominent textures are also captured. The equations (2.7) and (2.8) may be used to calculate the CS-LBP characteristics.

$$CS-LBP_{p,r,t}=\sum_{i=0}^{N/2-1}S\left(\left|g_i-g_{i+N/2}\right|\right)2^i \qquad (2.7)$$

$$S(x)\begin{cases}1 & if\ x\geq t\\ 0 & otherwise.\end{cases} \qquad (2.8)$$

Where g_i and $g_{i+N/2}$ equate to the grayscale of N centrally symmetric pairs of pixels, distributed evenly around a circle of radius r.

And N is the number of pixels other than the centre pixel; for a 3 × 3 region, for instance, N = 8. T is a less-than-positive number that is used to increase the CSLBP operator's resilience while smoothing the difference between grayscale images [43].

In the Figure (2.4), the threshold T is set to 4, and then the gray difference of four pairs of center symmetric pixels in the 3×3 window area is compared. When the threshold is less than T, the corresponding position will be set to 0, otherwise 1 will be set, and the CSLBP code value will be 0110.

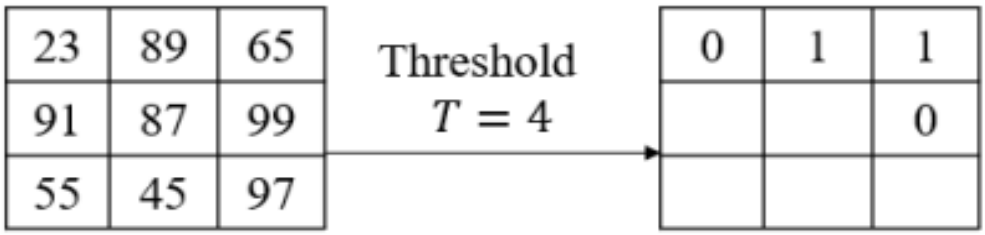


Figure (2.4): An example of the CS-LBP [44]

The feature dimension extracted by CS-LBP operator is 2^4=16, substantially less than that retrieved by the LBP operator, which significantly shrinks the feature dimension during the statistical histogram procedure and accomplishes the reduction goal. information is obtained in a gradient direction and stored and computed [44].

- **Extended Center-Symmetric Local Binary Pattern (XCS-LBP)**

The Extended Center-Symmetric Local Binary Pattern (XCS-LBP) descriptor is used in videos. It seems to be resilient to noise and variations in light since it combines the best aspects of the original LBP and related CS-LBP techniques. It also provides short histograms [45].

That's the CS-LBP extension factor by comparing the gray values of pairs of identical central pixels so that the resulting histogram also short, but also taking into account the central pixel. This set makes the resulting descriptor lower noise sensitive, called XCS-LBP (Extended CS-LBP), It is expressed by equation (2.9):

$$XCS - LBP_{p,r,t} = \sum_{i=0}^{N/2-1} S\big(g_1(i,c) + g_2(i,c)\big)2^i \qquad (2.9)$$

Where the threshold function s, which is used to determine the types of local pattern transition, is defined as a characteristic function (2.10):

$$s(x_1 + x_2) = \begin{cases} 1 & if(x_1 + x_2) \geq 0 \\ 0 & otherwise \end{cases} \quad (2.10)$$

and where $g_1(i, c)$ and $g_2(i, c)$ are defined by (2.11):

$$\begin{cases} g_1(i,c) = (g_i - g_{i+(p/2)}) + g_c \\ g_2(i,c) = (g_i - g_c)(g_{i+(p/2)} - g_c) \end{cases} \quad (2.11)$$

The XCS-LBP account is shown in Figure (2.5)

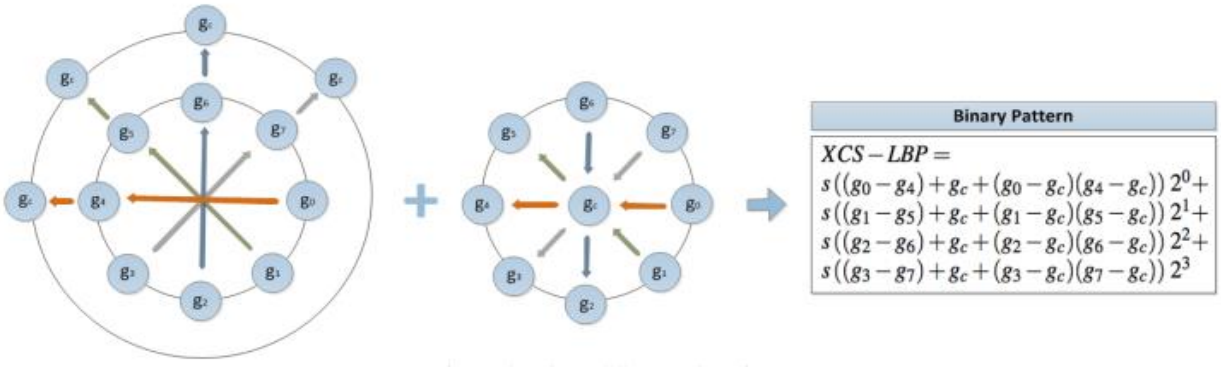


Figure (2.5): The XCS-LBP descriptor [45]

2.6 Machine Learning Models (ML)

A subset of artificial intelligence known as "machine learning" refers to the techniques and algorithms that learn from data rather than having the rules explicitly written by a domain expert, as was done during the height of expert systems the 1980s. In many cases, ML algorithms outperform manually designed algorithms in basic issues and provide answers to complicated problems like computer vision. ML algorithms uncover the significant patterns in the data on their own [46].

The study area known as "machine learning" aims to enable computers to learn without explicit programming. ML is used to train computers to process data more effectively. Sometimes, even after examining the data, we are unable to understand the knowledge gleaned from it. We utilize ML in this situation. The availability of numerous data sets has increased the need for ML. Machine learning is being used by many sectors to retrieve pertinent data [47].

Machine learning algorithms are trained on a large amount of data to learn patterns, relationships, and statistical models that can be used to make predictions or take actions on new unseen data.

The four basic types of machine learning algorithms are supervised, unsupervised, semi-supervised and reinforcement. These categories are based on how the algorithms handle the set of provided data and information. Although standard machine learning and detection strategies for abnormal walking behavior have provided relatively satisfactory results in past years, these approaches are usually limited by handcrafted features and limited ability to learn intrinsic patterns in the data [17].

2.7 Deep Learning Models (DL)

Deep Learning (DL) refers to a class of machine learning techniques and architectures that are characterized by the use of multiple hidden layers as shown in Figure (2.6) of nonlinear information processing stages, which are of a hierarchical nature. These architectures are designed to automatically learn and extract meaningful representations or features from raw data, without the need for explicit feature extraction [48].

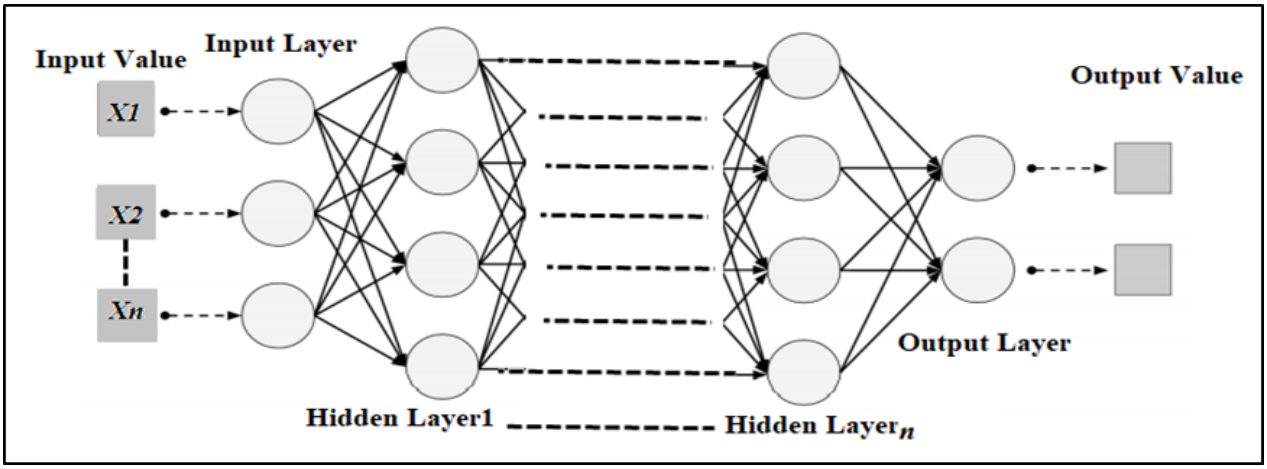


Figure (2.6): Deep neural network layers [49]

deep learning approaches emerged as an elegant solution for tackling image/video and sequential problems, among others, and a powerful tool for detecting abnormal gait behavior [17].

The ability of DL technologies to handle huge amounts of unlabeled data has made them powerful tools for handling big data analysis. DL uses several different types of neural networks, and the convolutional neural network (CNN) is one of the most important. DL can be categorized into three main groups: unsupervised (such as Autoencoder (AE), deep belief network (DBN), and generative adversarial network (GAN)), supervised (such as deep neural network (DNN), convolutional neural network (CNN), and recurrent neural network (RNN), and other hybrid techniques [50].

2.8 Convolutional Neural Network (CNN)

CNN is One of the most popular deep neural network types in computer vision applications that consists of multiple numbers of layers. It is a useful tool for object detection, image visualization, and image recognition. CNNs are the most widely used neural networks in AI when it comes to Deep Learning and image processing

since they process a lot of data, do not require manual feature extraction, and do not require complex segmentation [51].

The fundamental idea behind CNN is to merge local features from higher-layer inputs (often images) into lower-layer features that are more complicated. However, because of its multilayered construction, it requires a considerable amount of compute to train such networks [52].

Figure (2.7) depicts the typical CNN design, which consists of three different layers, CNN model is constructed consist from these layers (Convolution layer, Max pooling layer, Fully Connected Layer).

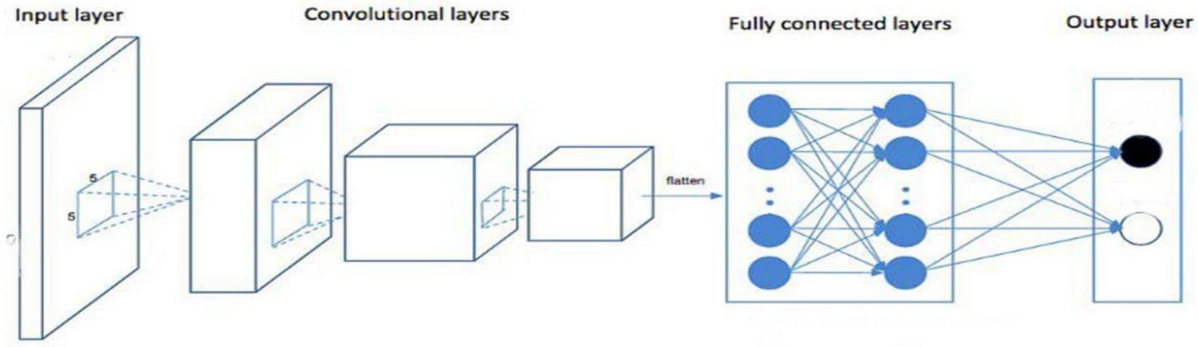


Figure (2.7): The architecture of convolutional neural network [53]

2.8.1 The Convolutional Layer

In the CNN design, the convolution layer is a fundamental layer [54]. It serves as the main structural component of the convolutional network that extracts feature from an input picture. A mathematical technique called convolutional combines two groupings of data. In order to produce feature maps, convolution filters are used to extract features from the input picture. These features are then learned using input data arrays. The filter was moved over an input picture to perform the convolution. The element-wise array is multiplied and the output is collected for each location point.

The feature map will then display this result. Example of convolution operation on a 5 x 5 input picture that is described as a 3D array with dimensions of "height, width, and depth" is shown in figure (2.8) RGB channels and the 3 x 3 filter equate to depth [55]

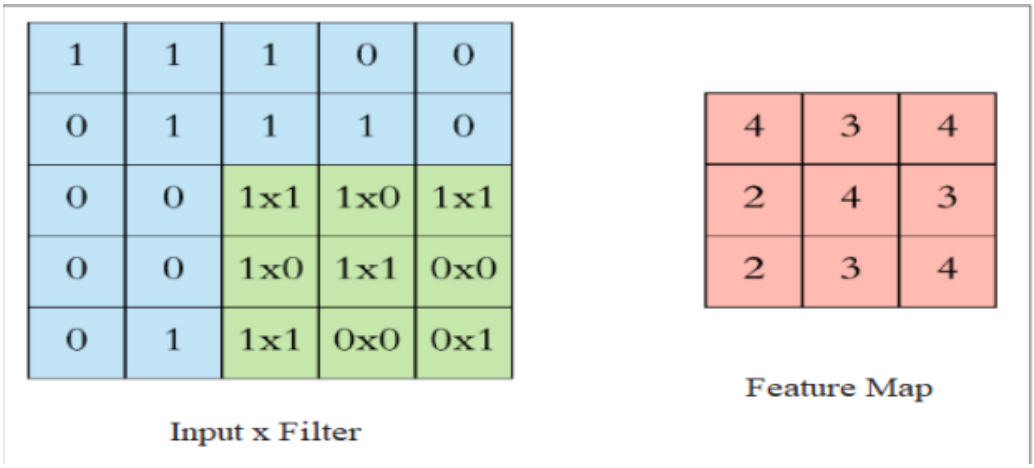


Figure (2.8): Convolution step with input image 5 x 5 and filter with 3 x 3 [55]

2.8.2 Max pooling layer

In order to reduce the number of parameters in the final model and to prevent overfitting during the training phase, the pooling layer reduces the size of the feature map, which is the output of the convolution network's final layer. Commonly, pooling layers with a down sampling ratio of 2 are positioned after the convolutional layer. It is a down sampling process that reduces the (height and width) of the map feature individually while leaving the depth untouched. It is applied to reduce the amount of noise in a picture. The two types of pooling most frequently used are average pooling and maximum pooling.

While the most common type of polling is max-pooling, which only chooses the maximum value in the pooling window, the average pool takes the average value from the input pixels [56]. Figure (2.9) illustrates different Pooling Methods.

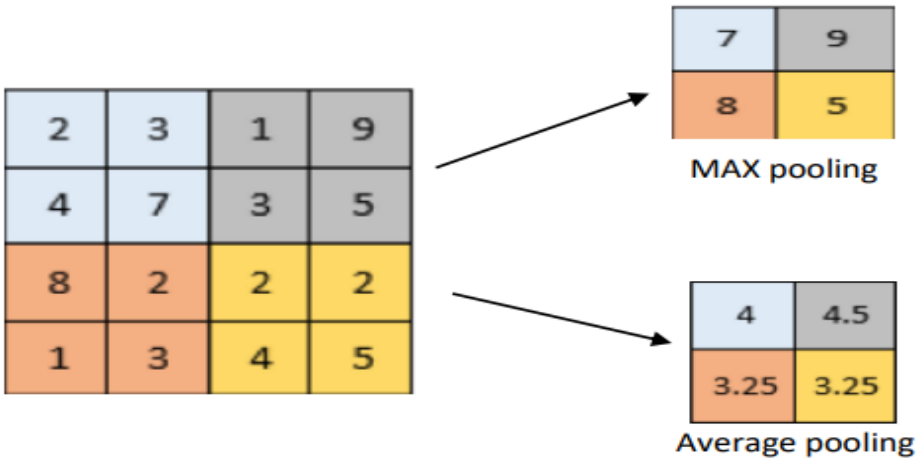


Figure (2.9): Different Pooling Methods [54]

2.8.3 Fully Connected Layer

The flattening procedure turns all of the two-dimensional arrays produced by the convolution and pooling layers from the pooled feature map into a single continuous linear vector. To categorize the image, the fully connected layer received as input a pixel from the flattened matrix.

To complete the CNN design, fully connected layers (FC) are included. The nodes that were obtained after the flattening process are connected by this layer, which serves as an input layer for the FC layers. Its objective is to use the features to categorize an input picture into several categories based on training data. It will be situated between the input layer and the output layer. Dense is also employed in the CNN building process known as the FC of the neural network. Additionally, a loss function is present in the last layer of fully connected models for model optimization and enhancing prediction accuracy [57]. Figure (2.10) show fully connected layer.

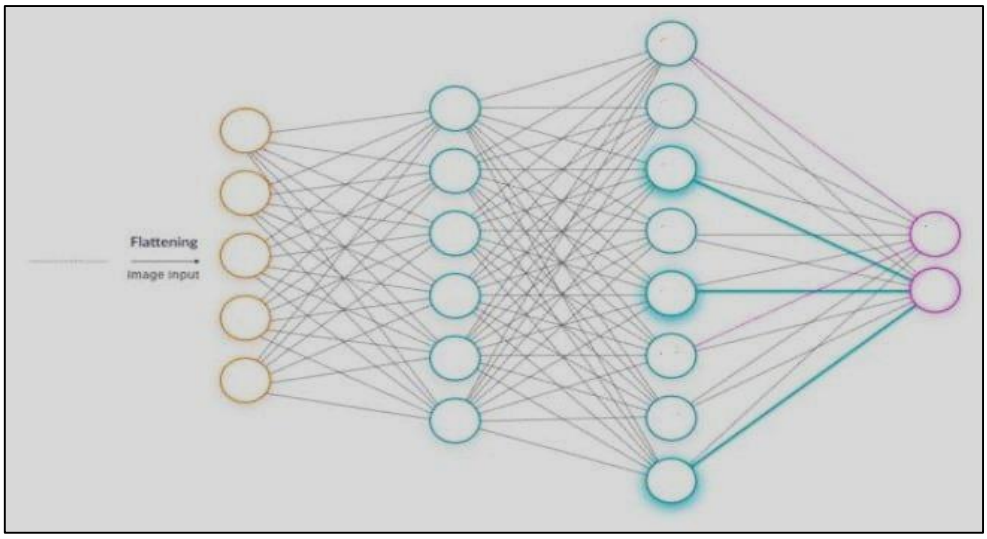


Figure (2.10): Fully Connected Layer [58]

2.9 Visual Geometry Group 16 (VGG16)

VGG16 (Visual Geometry Group 16) is a convolutional neural network (CNN) architecture that was proposed by the Visual Geometry Group at the University of Oxford. It was introduced in the paper titled "Very Deep Convolutional Networks for Large-Scale Image Recognition" by Karen Simonyan and Andrew Zisserman in 2014 [59]. The characteristics of the data set are extracted using the VGG16 layered CNN model. The layered architecture of the VGG16 is shown in Figure (2.11) It comprises of four fully connected layers, four maximum pooling layers, twelve convolutional layers, and a softmax classifier [60].

VGG16 has achieved excellent performance on various benchmark image classification tasks, such as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014. Its architecture has also served as a basis for subsequent CNN models, inspiring the development of deeper and more sophisticated networks.

Using the initial layers of the VGG16 architecture for feature extraction is a common technique in computer vision and deep learning tasks. VGG16 is a popular convolutional neural network (CNN) architecture that was trained on the ImageNet

dataset for image classification. It consists of multiple convolutional layers followed by fully connected layers.

When you use the initial layers of VGG16 for feature extraction, the main benefit of filters and feature maps from these layers to extract meaningful features from dataset without the need to train the entire network from scratch. This is known as transfer learning, and it can save you a significant amount of time and computational resources.

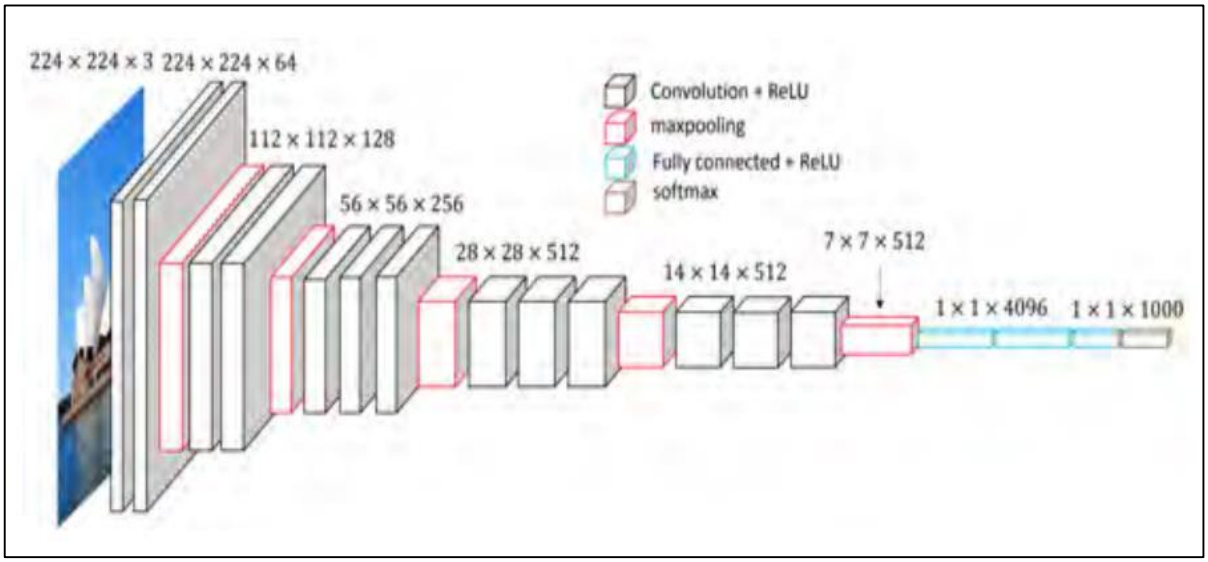


Figure (2.11): VGG16-CNN-Architecture [61]

2.10 The Fundamental Principles in Neural Networks

There are many key concepts that play a crucial role in shaping the network's performance: Loss functions, Activation functions, Optimizers and Normalization. These concepts contribute to the training and optimization of neural networks, allowing them to make accurate predictions and improve their overall performance. Below are details of these concepts:

2.10.1 Loss Function

Early neural network models compute error using the difference between the actual output and expected output. Numerous features of loss functions have recently been created to calculate error in neural networks. Since various loss functions may produce varying error levels for the same prediction, the type of loss function substantially influences the network's output. Loss functions can be categorized into one of three groups: Loss functions include, but are not limited to, embedding, regression, and classification loss functions.

Classification issues are addressed with classification loss functions. The Regression Loss functions are used in regression problems because the output variables are continuous. Embedding loss functions are used to quantify the similarity between two inputs.

- **Mean Square Error Loss Function (MSE)**

One of the most important regression loss functions, MSE, measures the square difference between the real and expected value [62]. The equation (2.12) displays the mean square error loss.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - p_i)^2 \qquad (2.12)$$

Where the $\mathbf{p_i}$ output forecast vector, $\mathbf{y_i}$ actual target vector, and **n** vector length are present.

- **Binary Cross-Entropy**

When a situation calls for binary classification, this strategy is employed by default and is the most effective one. When the goal values are either 0 or 1, it is presumptive and functional. The calculated value encapsulates the mean discrepancy between the calculated value probability distribution and the expected value probability distribution [63]. Zero is the ideal value for this type of method.

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \,.\log\big(p(y_i)\big) + (1 - y_i)\,.\log(1 - p(y_i)) \qquad (2.13)$$

where y_i is the label, p(y) is the predicted probability.

- **Categorical Cross-Entropy**

This loss function is employed in multi-class classification problems. In some cases, the target may only belong to one of several possible categories, and the model must determine which one. This function is widely used to compare two probability distributions since it is employed in networks that employ the Soft-max activation function [64].

The categorical cross-entropy is demonstrated by the following equation (2.14).

$$Lcross - entropy(y.y) = -\sum_{\iota=1}^{n'} y_\iota \log(\acute{y_{\iota)}} \qquad (2.14)$$

Where $\mathbf{y_i}$ actual target vector, $\tilde{\mathbf{y}}_i$ the output predicted vector, **i** vector length.

- **Euclidean Distance Loss**

The loss function is shown here. Instead of categorization concerns, situations where two inputs must be compared are where embedding loss is most frequently used. It determines the separation between two points or vectors. The following equation (2.15), which represents the Euclidean Distance Loss [65].

$$Euclidean\ loss = \sqrt{\sum_{i=1}^{n}(y_i - p_i)^2} \qquad (2.15)$$

Where $\mathbf{y_i}$ is the actual input vector and **n** is the vector's length, $\mathbf{p_i}$ is the projected vector.

2.10.2 Activation Function

The activation functions are used to produce the output of the neural network. Activation functions are available in a variety of shapes and sizes depending on the task at hand. The two primary kinds of activation functions are linear and nonlinear. Normal ranges are either (-1,1) or (0,1). It has significance to achieve non-linearity in a multilayer neural network in order to make it non-linear. The importance of a

non-linear activation function in a neural network might be attributed to the existence of data that cannot be partitioned in a linear fashion. Because of this, using merely a linear activation function does not allow a multi-layer or deep neural network to benefit from the additional layer. Because it may translate the output of a network into a constrained range, nonlinear activation functions are essential for particular networks. The most common non-linear activation functions used in neural networks are sigmoid, tanh, Soft-max, and ReLU [66][67].

The standard option for the activation function in hidden layers is the (sigmoid) function, which is the activation function for probability calculations using the equation formula (2.16).

$$g(x) = \frac{1}{1+e^{-x}} \qquad (2.16)$$

The rectified linear unit (ReLU), uses a nonlinear pixel-wise function called g. It returns x if x is positive; else, it returns 0, using the equation formula (2.17)

$$g(x) = \max(x, 0) \qquad (2.17)$$

2.10.3 Optimizer

optimizer is an algorithm that adjusts the attributes of the neural network, such as weights and learning rates. Changing and adjusting various movable factors. Finding the best values for these variables to reduce or maximize the output of the function in the big and high-dimensional space is a challenging. In reality, the network's weights and biases are the variables that the optimizer regulates. There are several varieties of optimizers such as (Adam, Ada Grad, SGD) [68].

It is used to determine the network weight parameters, the Adam (Adaptive Moments) optimizer is employed. By computing the gradient's first-order moment prediction and second-order moment prediction, independent adaptive learning rates

are generated for a number of parameters. According to empirical findings, Adam outperforms other optimizers in real-world settings [69]. Moving gradient averages

$$m_t = \beta_1 m_t - 1 + (1 - \beta_1) g_t \quad (2.18)$$

And squared gradient:

$$v_t = \beta_2 v_t - 1 + (1 - \beta_2) g^2_t. \quad (2.19)$$

Where m_t: The exponentially weighted moving average at time *t*, β_1: The smoothing factor, where $0<\beta 1<1$, $m_t - 1$: The exponentially weighted moving average at the previous time step, g_t: The gradient at time *t*.

2.10.4 Normalization

One of the crucial preprocessing procedures is normalizing the training and testing datasets. To reduce the impact, training sets' varied features are scaled numerically. Through normalization, the value is scaled to [0, 1], and the normalized value x' is

$$\acute{X} = (X - X_{min})/(X_{max} - X_{min}) \quad (2.20)$$

Where *X* is the starting value, X_{max} and X_{min} are the features heights and lowest values. The normalization of data is demonstrated by the algorithm (2.1).

Algorithm (2.1): Normalization
Input: input image **Output**: : normalized image
begin **step1:** for i = 1 to x do **step2:** for j= 1 to x do **step3:** find the minimum value of x (min_i) **step4:** find the maximum value of x (max_i)

step5: $x_{ij} = \frac{x_{ij} - min_i}{max_i - min_i}$

step6: end for

step7: end for

End

2.11 Performance Evaluation Metrics

There are several common metrics, including accuracy, precision, recall, and F-measure, that may be used to compute and assess the performance of the proposed system. we will need to make use of a matrix of computing confusion to figure out the computations for these metrics. The classification model performance matrix is one of the most important tools for describing the performance of the model. Table (2.1) displays the confusion matrix for a binary classification model.

Table (2.1): Confusion matrix

	Detected	
Actual	**Normal**	**abnormal**
Normal	TP	FP
abnormal	FN	TN

- True positive (**TP**): refers to occurrences of positivity that are categorized properly.
- False Negative (**FN**): These are negative occurrences that have been categorized erroneously.
- False Positive (**FP**): These are positive occurrences that are categorized erroneously.
- True negative (**TN**): refers to occurrences of negativity that are identified properly.

The methods for computing each of these metrics are described in the following:

- **Accuracy (ACC):** Is the typical accurate prediction. This is determined by dividing the accurate prediction by the total number of forecasts. This will give the entire network a single value. show in equation (2.21)

$$ACC = (TP + TN)/(TP + TN + FP + FN) \quad (2.21)$$

- **Recall:** The True Positive rate, which represents the completeness of a model. Which is calculated by dividing the correctly detected phenomena (True Positive) over total true cases of that phenomenon (True Positive + False Negative).

$$Recall = TP/(TP + FN) \quad (2.22)$$

- **Precision:** Indicate how accurate the learned model is. Which tells us how many of the identified positive instances are actually positive.

$$Precision = TP/(TP + FP) \quad (2.23)$$

- **F1-score:** Displays a combination of sensitivity and accuracy for calculating a balanced mean output.

$$Fscore = 2 * ((Precision * Recall)/(Precision + Recall)) \quad (2.24)$$

Chapter Three

Proposed System

Chapter Three

Proposed System

3.1 Introduction

In this chapter, a detailed description of the proposed system will be provided, together with an explanation of its overall structure and all of the algorithms employed in its construction. The proposed system consists of two main phases: the training phase and the testing phase. In the training phase, the system is trained on a labeled dataset to learn the patterns and characteristics of normal and abnormal behavior in images. The testing phase is the process of applying the trained system to unseen images to classify them as normal or abnormal.

The first step in both phases is pre-processing, which involves several processes to prepare the video data for further analysis. The video is divided into a sequential set of images, which allows the system to analyze each frame individually. The obtained images are then converted to grayscale, which simplifies the analysis by removing color information. The images are resized, ensuring consistency in the input data. Noise removal techniques are applied to enhance the quality of the images, reducing any unwanted artifacts or disturbances. after which the data is normalized.

The next step is the feature extraction. In this step, a set of features is extracted from each image to capture the relevant information for classification. Several feature extraction methods are used (HOG, LBP, CS_LBP, XCS-LBP), these methods are chosen to explore different aspects of the image data and find the most discriminative features for the classification task.

After feature extraction, the final stage is the classification. The extracted and selected features are used as input to a (CNN-Vgg16) classifier. The output of the classifier is a prediction of whether the gait pattern in the video is normal or abnormal. To evaluate the performance of the system, used common evaluation metrics such as accuracy, precision, recall, and F1-score.

3.2 The Proposed System

As it is known, the architecture of any proposed system is crucial because it explains how that system works. The system functions and details all procedures and phases in detail. It is done to get the needed information from the system. Based on the video stream that was recorded for that individual and stored in a file, the suggested system may identify a person's behavior. However, the major goal of the proposed system is to classify a person's gait from a video based on the manner in which he moves and to classify whether such gait pattern is normal or abnormal.

all frames from recorded video clips will be stored to a file for image data collection, which will then be used to create a performance classifier. Figure (3.1) shows the main stages of the proposed system for the process of building a sequential classifier and how it was used to detect a person's behavior based on his gait. phases of pre-processing data to improve the datasets suitability for the proposed classifier. Creating a new model to extract the characteristics of the task from the data, running and analyzing the experiment at this point, comparing the suggested model to the most recent approaches, using a variety of performance criteria, and finally confirming the findings using evaluation metrics.

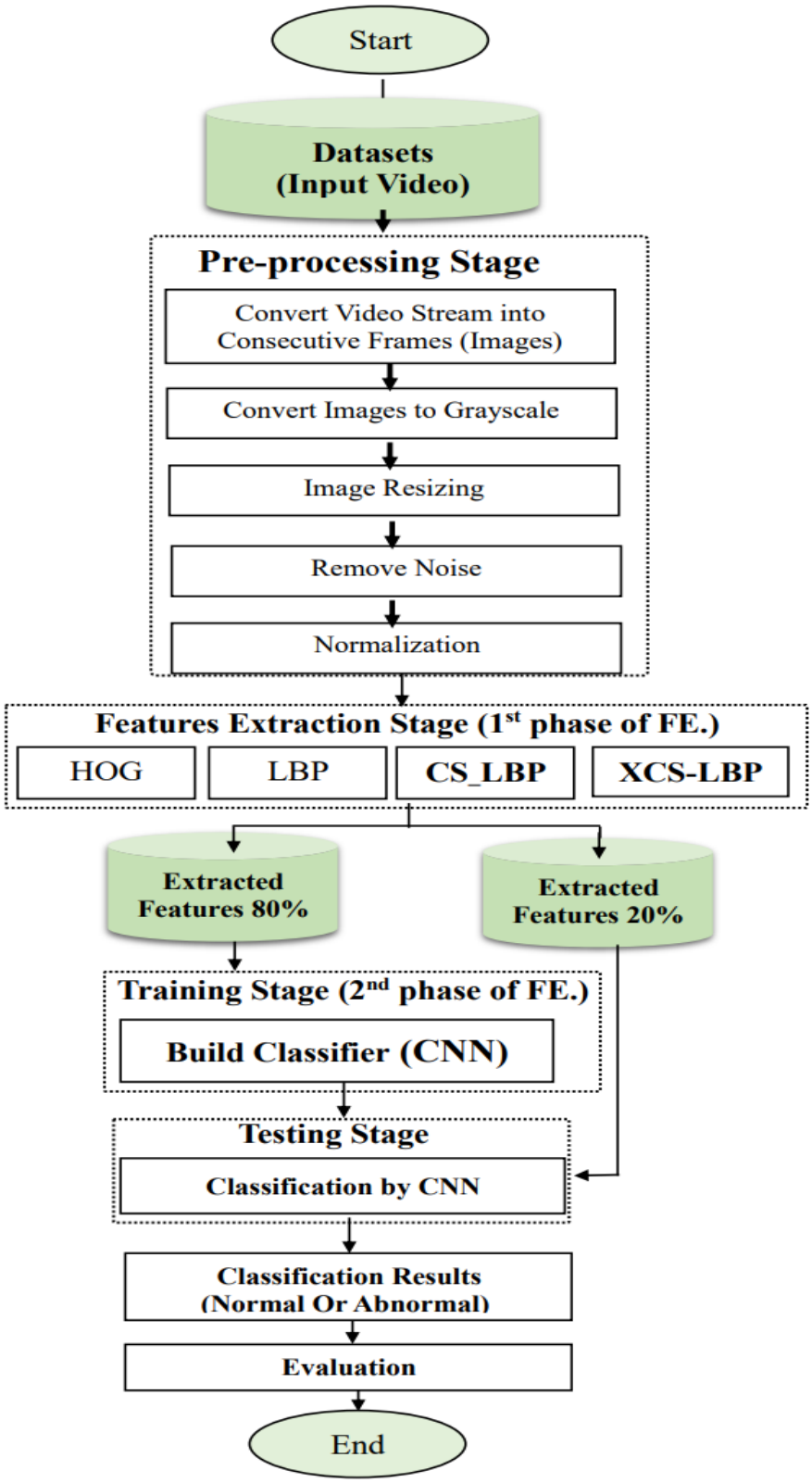


Figure (3.1): General Block Diagram of the Proposed System

3.2.1 Dataset

Creating a dataset with labeled examples of both normal and aberrant gaits is indeed a crucial step in training AI models for gait recognition or abnormality detection. Having a diverse dataset that covers a wide range of healthy and abnormal gaits helps the model learn the characteristics and patterns associated with different movement patterns. With this dataset, you can use deep learning techniques, to train AI models that can identify and classify gait as normal or abnormal. By feeding the model the labeled dataset, it can learn to extract relevant features and patterns from the gait and make predictions from the unseen data.

Having a quality dataset is a key factor in training effective models. To ensure the reliability and accuracy of the system's models, carefully curated to accurately classify gaits. The collection of the dataset was highly restricted due to the lack of task-specific training data. The dataset consists of a collection of videos portraying the walking patterns of two distinct groups: healthy individuals with 'normal' gaits and patients diagnosed with various medical conditions that lead to 'abnormal' gaits. The video contains one person walking, either natural or unnatural. Labeling is performed for the videos and a (label = 1) is given to abnormal videos and (label = 0) to normal videos.

The data collection process was conducted manually, involving the careful selection and extraction of videos from online sources. These sources were accessible through web browsing platforms such as (https://youtu.be/7SyTpEdhBLw?si=v6y5vVq _cRT_R7JM) and provided videos demonstrating individuals' gaits in different scenarios. This dataset consisted of 112 videos, which were categorized into 53 normal recordings and 59 abnormal recordings. After dividing these videos into frames in the pre-processing stage, we got 4909 frames. The frame selection of the

video was within a specific ratio, where each second had a (frame-rate =0.2). This data collection is briefly described in Table (3.1) and Figure (3.2) that displays portion of the dataset used.

Table (3.1): Briefly described of the dataset used.

Class Type	Number of videos of all class	Number of frames taken from all videos
Normal(0)	53	1710
Abnormal(1)	59	3199
Total	112	4909

Figure (3.2): Top row containing four frames show abnormal gait patterns and frames in bottom row show normal gait patterns.

In this proposed system, training was done using 80% of the dataset. While the remaining 20% of the dataset was used to test the model once it had been trained.

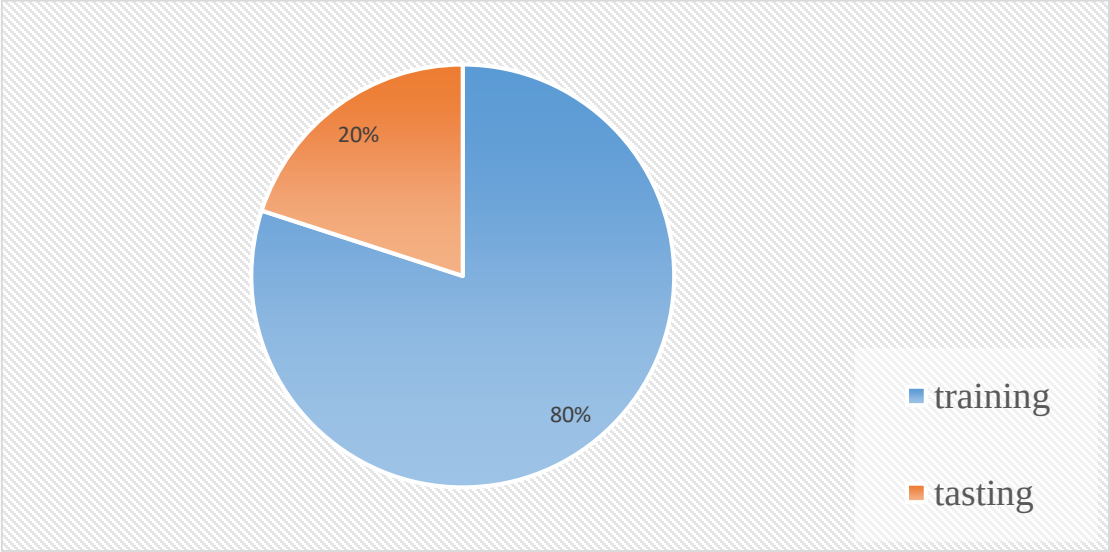


Figure (3.3): The ratio of the training dataset to the testing dataset.

3.2.2 Pre-processing Stage

Pre-processing is the process of applying a set of prior actions to an image (frame) in order to prepare it for the following stage. The fundamental benefit of pre-processing is that it organizes the data and makes the filtering process easier. Preprocessing is done to simplify dataset to subsequent step, features extraction. The steps implemented on the dataset during the pre-processing stage are represented by steps below:

3.2.2.1 Convert Video Stream into Consecutive Frames.

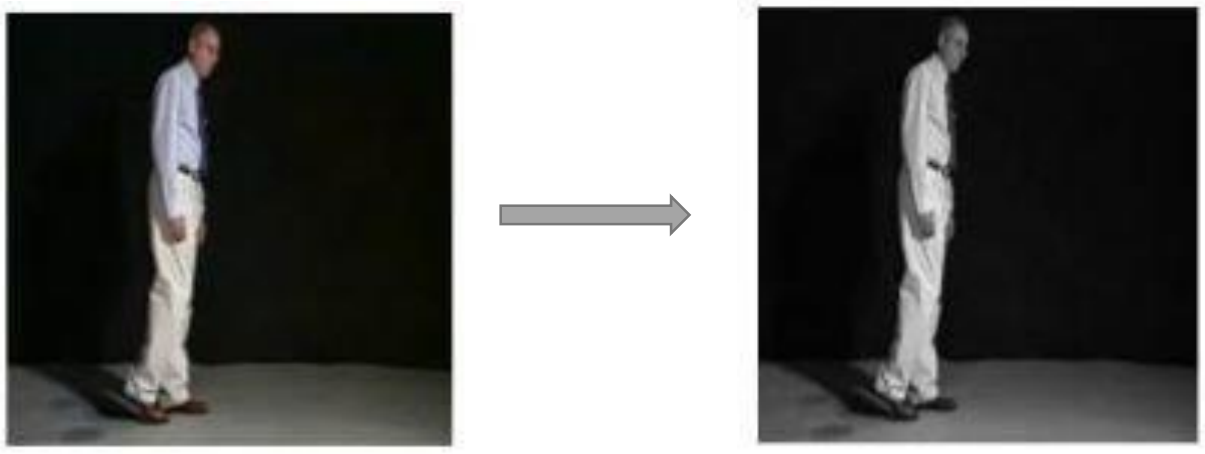
A person is recognized based on their gait pattern, which was captured from a video motion previously taken with a digital camera. The video package is handled by a number of different processes after being broken down into a collection of pictures (known as frames).

3.2.2.2 Images Converting to Grayscale

All frames created from video need to be prepared and improved by deleting any extraneous information in order to be ready for the following step. These images, represented by three-dimensional RGB bands, need to be changed to grayscale in order to decrease the quantity of channels from three to one in the image graph facts entered. Enter the grayscale picture, using grayscale photographs speeds up and improves the efficiency of the verification process. seen in Figure (3.4).

The typical gray-scale intensity, which is represented as an 8-bit integer, has 256 possible shades of grey, from white to black. The conversion to the gray level is as in the following equation (3.1)

$$gray=(0.299\times red) +(0.587\times green) +(0.114\times blue) \quad (3.1)$$



(a) **(b)**

Figure (3.4): Images converting to Grayscale (a) RGB image (b) Grayscale image

3.2.2.3 Image Resizing

A uniform (standard) size can be specified for all images submitted to selected AI algorithms due to the size of most images captured by the camera and submitted to a variety of selected AI algorithm. The images have been reduced to (244 x 244) pixels. It is important to note that the choice of image size depends on the specific requirements and limitations of the algorithms or systems involved. The size of (244 x 244)pixels may be adequate for some image recognition or classification tasks, especially when working with restricted computational resources or when dealing with smaller images. This size can be useful especially in our proposed system because we used images with of 244 x 244 pixels to correspond to a specific CNN architecture such as VGG16 which often requires a fixed input size to function effectively, larger image sizes lead to higher computational requirements during training and inference due to the increased number of pixels. By choosing a reasonable image size such as 244 x 244 pixels, we will strike a balance between preserving essential image detail and ensuring computational efficiency. This allows the model to be trained more efficiently.

3.2.2.4 Noise Removal (Remove Noise)

Noise is added during image collecting and transmission. The crucial and key phase in the idea of image processing is picture optimization. It is used to enhance the brightness and clarity of the photographs as well as lessen their noise. During this procedure, the images undergo cleaning to remove standard recognized noises and imperfections. Image noise, characterized by variations in brightness and color intensity among the image's pixels, often occurs during the image capture or scanning process, induced by artificial sources like flashlights or natural factors like strong sunlight.

3.2.2.5 Normalization

It is a crucial stage in the pre-processing of images. This is as a result of the [0-1] range of pictures that CNN receives and analyses. To get the desired outcome, each pixel in the [0-255] to [0-1] range is rescaled by being divided by 255.
The minimum () and maximum () values are changed to 0 and 255 when working with an 8-bit image. When the value of minimum () gets close to 0, the simple math to convert an 8-bit image into a floating-point image is (image/255).

3.2.3 Feature Extraction

In order to provide deep learning the best chance of understanding the visual pattern, millions of numerical pixel values are encoded as feature vectors. The most important element of any pattern classification system is feature extraction; the more distinct features are extracted, the easier the classification task becomes. How to pick out and extract the key characteristics that help us get the right categorization results after optimizing the image and making it a grayscale image. System accuracy and calculation time are impacted by irrelevant feature data collected from the original frames.

In the proposed system, several methods were applied for the purpose of obtaining the extracted features useful for the classification process, and they were initially applied to one frame of each video, so that the total dataset reached 112 frames, and again more than one frame was taken for one video, and the total frames were 4909 frames. Below we will mention all the methods used to extract features in our proposed system:

A. Without Feature Extraction Method

In this case, the dataset enters directly to the classification stage, that is, raw dataset is entered only, applying the pre-processing stages. The dataset in this case is without extracted features, as in the algorithm (3.1).

Algorithm (3.1): Without Feature Extraction Method
Input: dataset (DS) **Output**: Type of classification (normal , abnormal)
Begin **Step1:** Read DS **Step2:** Apply pre-processing steps in DS **Step3:** Convert Video Stream into Consecutive Frames. **Step4**: Images converting to Grayscale. **Step5:** Image Resizing to 244×244 grayscale 8-bit image. **Step6:** Noise removal (Remove Noise). **Step7:** Normalization **Step8:** Apply (CNN+Vgg16) classifier with training part of the dataset. **Step9:** Use the CNN classifier to Return type of classification (normal, abnormal) of the testing part of the dataset. **Step 10:** Results Evaluation. **End.**

B. Feature Extraction based on HOG Method

Histogram of oriented gradients, or HOG, is a feature descriptor. The HOG descriptor can be used in computer vision or image processing for object detection. The process determines gradient orientation. Occurrences in a certain area of a photograph. The HOG focuses on the structure or form of an item. It determines the features by using both the gradient's magnitude and angle.

We figure out the gradient pictures. The gradients in the horizontal and vertical directions must first be determined; Ultimately, the goal is to compute the gradient graph. This can be easily achieved by filtering the image using a sobel mask to get Gx and Gy, after which the magnitude and direction of gradient are calculated. Algorithm (3.2) illustrates the feature extraction process using HOG descriptor.

Algorithm (3.2): Feature Extraction based on HOG Method
Input: Dataset (DS) // Dataset after apply pre-processing. **Output**: Image features extracted .
Begin **Step1:** Image features extracted = [] // empty list **Step2**: Load DS **Step3**: for i =1 to length list of image: // Repeat for each frame in the list of image **Step4**: Calculate gradient of the image Gx and Gy by using equation (2.1) and (2.2) **Step5**: Calculate magnitude of the image by using equation (2.3). **Step6**: Calculate angle of the image by using equation (2.4). **Step7:** Appends the magnitude to list Image features extracted

Step8: end for

Step9: Return Image features extracted

End.

C. Feature Extraction based on LBP Method

Another method implemented is the LBP method, which is a feature descriptor. The LBP descriptor can be used in computer vision or image processing to detect the object, and the LBP feature vector is calculated by dividing the scanned window into 3 * 3 pixels per cell. Comparing each of the eight cells adjacent to each pixel (upper left, middle left, lower left, upper right, etc.). Circularly, either clockwise or counterclockwise, move the pixels. Type "0" in cases where the value of the central pixel is higher than the value of its neighbor. If not, type "1". This results in an 8-digit binary number, which is often easily translated into a decimal number.

Counting the frequency of each number (that is, each group of pixels smaller and larger than the center) occurring over the cell. Might think of this histogram as a 256-dimensional property vector. Can choose to normalize the graph. Sequential graphs of all cells. This gives a vector feature for the entire window. These steps were applied in the proposed system to obtain another way to extract the feature. This is explained with the Algorithm (3.3).

Algorithm (3.3): Feature Extraction based on LBP Method

Input: Dataset (DS) // Dataset after apply pre-processing.

Output: Image features extracted

Begin

Step1: Image features extracted **=** [] // empty list

Step2: for k =1 to length list of image: // Repeat for each frame in the list of image

Step3: image-gray = DS[k]

Step4: The **height**, **width**, and **number of channels** (assuming 1 for gray) are extracted from the image shape.

Step5: An empty LBP-image with the same height, width, and 1 channels as the original image

Step6: for i=1 to (0, height) // Nested loops are used to iterate over each pixel in the image.

Step7: for j=1 to (0, width)

Step8: LBP-image (i,j) =For each pixel, equations (**2.5**) and (**2.6**) are applied to calculate the LBP value, and the result is mapped to the according to pixel in the LBP image.

Step9: end for

Step10: end for

Step11: Image features extracted = appends the LBP-image to list Image features extracted

Step12: end for

Step13: Return Image features extracted

End

D. Feature Extraction based on CS-LBP Method

The CS-LBP approach, a feature descriptor, is another technique used. The CS-LBP descriptor can be used in image processing or computer vision to find the object; it is an updated method for LBP.

Using the CS-LBP operator, which was inspired by the local binary patterns LBP, a feature for each pixel is extracted from the region after preprocessing. It is challenging to employ the LBP operator in the context of an area descriptor since it generates quite lengthy histograms.

Only center-symmetrical pairs of pixels are compared in order to compare more compact binary patterns. We can observe that LBP creates (256) distinct binary patterns for 8 neighbors, but CS-LBP only produces (16). Additionally, by thresholding the gray level differences with a modest value T = 0.1, resilience on flat picture portions is produced. Algorithm (3.4) shows the working steps of the CS-LBP method:

Algorithm (3.4): Feature Extraction based on CS-LBP Method
Input: Dataset (DS) // Dataset after apply pre-processing. **Output**: Image features extracted
Begin **Step1:** Image features extracted = [] // empty list **Step2:** An empty CS-LBP-image with the same height, width, and 1 channels as the original image **Step3:** neighboor = 4 **Step4:** for i=1 to (0, DS [0] - neighboor) **Step5:** for j=1 to (0, DS [1] - neighboor)

Step6: image = DS (i + neighboor ,j+ neighboor)//Get matrix image 3 by 3 pixel

Step7: CS-LBP-image [i+1, j+1] =Depending on the eight neighbors in image, equations (2.7) and (2.8) are applied to find the value of the center

Step8: end for

Step9: end for

Step10: Image features extracted = appends the CS-LBP-image to list Image features extracted

Step11: Return Image features extracted

End

E. Feature Extraction based on XCS-LBP Method

A descriptor is called an eXtended Center-Symmetric Local Binary Pattern (XCS-LBP) for modeling and removing the backdrop from films. It appears to be resilient to variations in light and noise and also generates short histograms by fusing the advantages of the original LBP and related CS-LBP ones. Algorithm (3.5) shows the working steps of the XCS-LBP method:

Algorithm (3.5): Feature Extraction based on XCS-LBP Method
Input: Dataset (DS) // Dataset after apply pre-processing. **Output**: Image features extracted
Begin **Step1:** Image features extracted = [] // empty list **Step2:** An empty CS-LBP-image with the same height, width, and 1 channels as the original image

Step3: neighboor = 4

Step4: for i=1 to (0, DS [0] - neighboor)

Step5: for j=1 to (0, DS [1] - neighboor)

Step6: image = DS (i+ neighboor, j+ neighboor)//Get matrix image 3 by 3 pixel

Step7: XCS-LBP -image [i+1, j+1] =Depending on the eight neighbors in image, equations (2.9), (2.10) and (2.11) are applied to find the value of the center pixel.

Step8: end for

Step9: end for

Step10: Image features extracted = appends the XCS-LBP-image to list Image features extracted

Step11: Return Image features extracted

End

3.2.4 Classification Process

One of the most important classes is CNN. After pre-processing and feature extraction, it uses an input image, analyses it, and then categorizes it into particular classes. The CNN model is implemented using the Python-based deep learning package keras, which is free and open source. The layering in the suggested system is inserted after the input layer while creating a dense and multi-layered CNN, and each pre-processed image will travel through a succession of convolution layers with filters (kernels), pooling, and Fully Connected Composition (FC) layers for gait classify. Figure (3.5) displays The Structure of the CNN+Vgg16 in roposed system for the provided dataset.

Layers from Keras applications and weights from those learned on the ImageNet dataset were used to generate the pre-trained VGG16 model. The VGG16 model's layers are then all configured to be untrainable. The last four layers of VGG16 have been rendered untrainable. The network architecture was constructed using VGG16, and the numerous levels that were constructed were then tracked. Here, we used VGG16 to extract solely convolutional and pooling layer-specific unique features. It provides separate entries to the following layers and, by fusing this structure with the constructed CNN's structure, achieves a stronger categorization. Algorithm (3.6) illustrates Building a CNN structure.

Table (3.2): The Structure of the CNN+Vgg16 in Proposed System.

No.	Layer Name	Output layer	Param#
1.	input_4:Input Layer	(None,244,244,3)	/
2.	rescaling_2:Rescaling	(None,244,244,3)	/
3.	tf_operators_.getitem_1: slicingOpLambda	(None,244,244,3)	/
4.	tf.nn.bias_add_1:TFOpLambda	(None, 128, 128, 64)	/
5.	vgg16: Functional	(None,None,None,512)	14714688
6.	flatten_1: Flatten	(None,25088)	/
7.	dense_2: Dense	(None,250)	6272250
8.	dropout_1: Dropout	(None,250)	/
9.	dense_3: Dense	(None,1)	251
Total parameter: 20,987,189 Trainable parameters: 6,272,501 Non-trainable parameters: 14,714,688			

Algorithm (3.6): Building a CNN structure

Input: Image features extracted

Output: CNN Model

Begin

Step1: Create layers model

Step2: Inputs = the input shape is (244x244) pixels with will have three identical grayscale channels.

Step3: The input data is first rescaled by dividing every pixel by 255.

Step4: The preprocess input function from VGG16 is applied to the input, which preprocesses the input according to the requirements of the VGG16 model.

Step5: conv_base represents the convolutional base of a pre-trained model (VGG16). This assumes that conv_base is defined and performs the necessary convolutional operations on the input.

Step6: Build flatten layer, the extracted features from the previous layer are flattened into a 1-dimensional tensor.

Step7: Build fully connected (dense) layer with 250 neurons.

Step8: Build a dropout layer and apply a 0.5 rate to discard a portion of the input units at random during training to minimize overfitting.

Step9: Output = Build Dense layer with 1 neuron and a sigmoid activation function is used for binary classification.

Step10: model = Model (Inputs, Output)

Step11: model.compile(optimizer=adam, loss='binary_crossentropy', learning_rate=0.001, metrics=['accuracy']).

Step 12: Training the CNN model on the "training" partition (80% of the dataset). **Step13:** Save CNN model. **End**

3.2.5 Testing Phase

In the test phase of convolution neural network CNN, the test is performed on the unseen test data, this phase consists of the following steps:

1. Insert a test image from the test dataset, which is 20% of the dataset and randomly selected.
2. Perform preprocessing before feeding the test, and use (HOG, LBP, CS-LBP, or XCS-LBP) to collect features. Test the features of the images so that the trained classifier can decide.
3. When making a decision, the result will be either normal or abnormal behavior.

In the testing phase, unknown images are first processed by pre-processing and then input to the feature extraction step; convert the extracted features to classification step. Algorithm (3.7) general testing steps of proposed system**:**

Algorithm (3.7): Steps for General Testing the Proposed System
Input: Test video **Output**: Type of classification (normal , abnormal)
Begin **Step1:** Read Test video **Step2:** Apply pre-processing steps in Test video **Step3:** Convert Video Stream into Consecutive Frames.

Step4: Images converting to Grayscale.

Step5: Image Resizing to 244×244 grayscale 8-bit image.

Step6: Noise removal (Remove Noise).

Step7: Normalization

Step8: Selected feature by feature extraction method (HOG, LBP, CS_LBP, XCS-LBP)

Step9: Apply CNN classifiers

Step10: Return Type of classification (normal , abnormal).

End.

3.2.6 System Evaluation

Once the model has completed training, need to be evaluated its performance and determine whether it is suitable for using with the training data set. This must become the criterion for determining how well a model performs.

Performance of the model and its ability to generalize to new data that the model has not encountered before are evaluated in a stage called "evaluation". This is achieved by employing a number of metrics, the specifics of which are mentioned in Chapter 2: accuracy, recall, precision, and f1-score.

Chapter Four

Experimental Results and Discussions

Chapter Four

Experiments Results and Discussions

4.1 Introduction

In this chapter, the experimental results produced by the proposed system will be presented on a manually collected dataset as mentioned previously. The results showed the success of this work in discovering behavior based on human gait. The performance of the proposed method was measured based on metrics such as accuracy, precision, recall, and F1 score. We will discuss the results in detail by going through the stages of implementing the system and presenting the results by figures and tables. Describe the language and environment in which the proposed system is implemented.

4.2 Hardware and Software Specifications

The proposed system has been implemented by using python language version (3.9.12)/ ANACONDA / Jupyter environment, with processor 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz 2.30 GHz and 16.0 GB RAM, 64-bit operating system, x64-based processor and Windows 10 Pro.

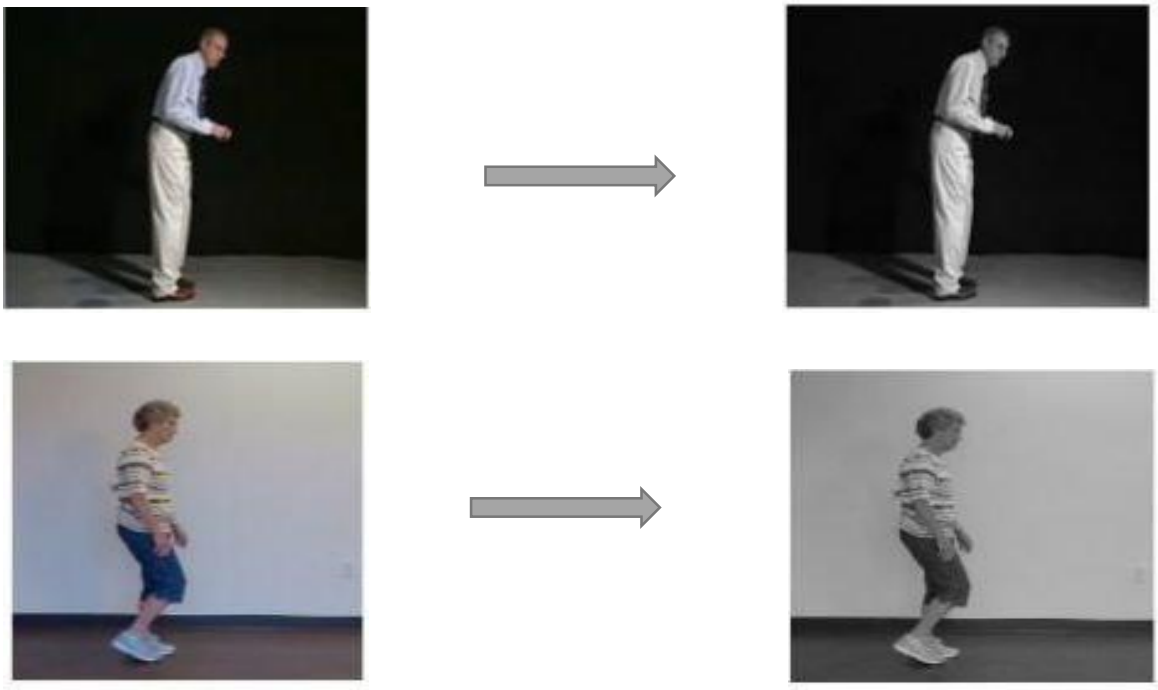
4.3 Experiments Results of the Proposed System

The dataset is used to train and test the proposed approach, and was collected manually. Video content was taken from different online website, these platforms provided access to a collection of videos, filming various walking scenarios of sick and healthy people. The dataset consists of 112 videos, 53 of which depict

individuals exhibiting “normal” gait patterns and 59 of which depict Individuals with “abnormal” gait patterns.

After that, the collected dataset goes through a pre-processing stage before applying one of the feature extraction methods. It goes through several steps in this processing, which are the step of converting the video into images, the step of converting these images to the gray level, and then the step of resizing the images, where the dimensions of each image are standardized to a resolution 244*244 pixels, and after that comes the step of removing the noise inside the images to enhance clarity and finally, a normalization process is applied to the data set.

Collecting and pre-processing a dataset is an essential step in training and evaluating machine learning models. Manual collection involves collecting the necessary data from different sources. Figure (4.1) shows the results of some of the images that has been obtained after the preprocessing stage:



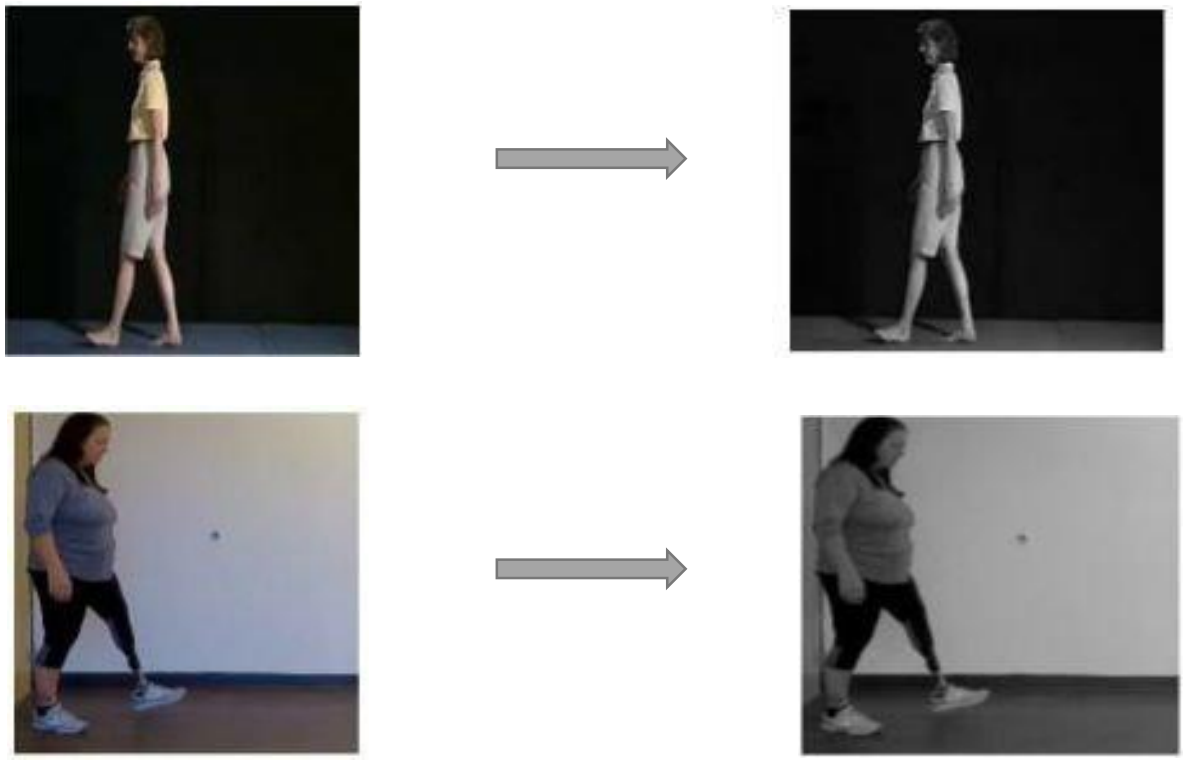


Figure (4.1): images obtained after the pre-processing stage

4.3.1 Results of Feature Extraction

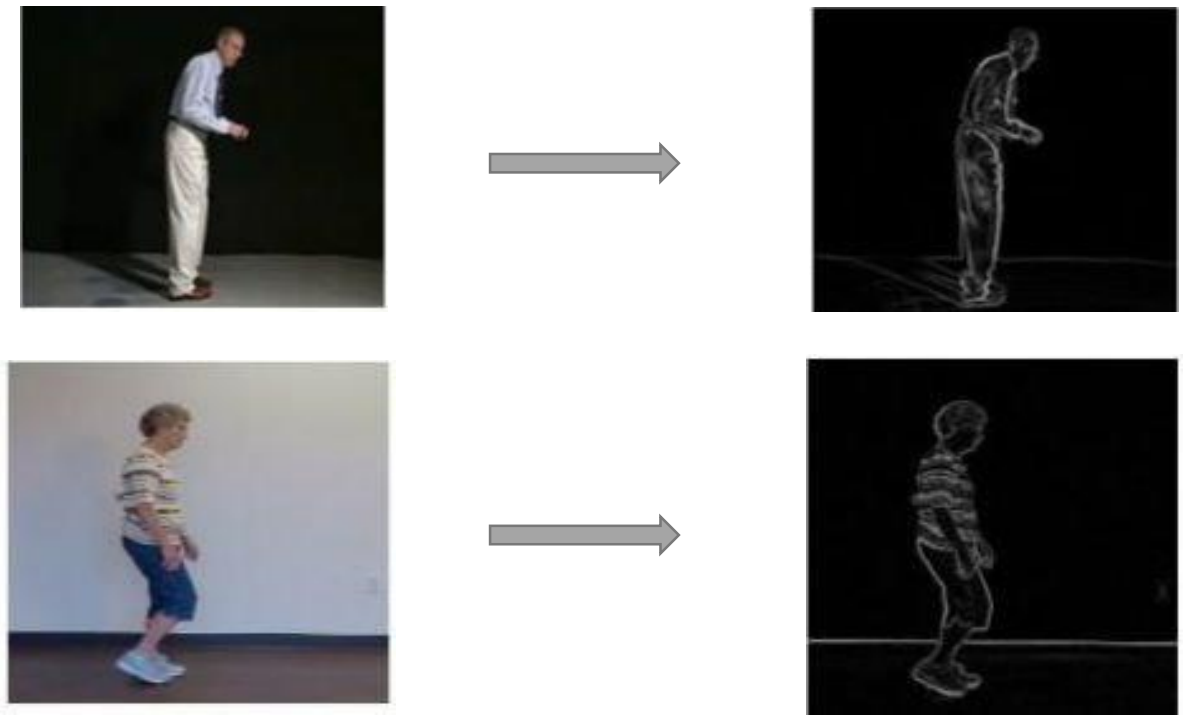
After the dataset was pre-processed and obtained the previous results. A new stage begins by applying several feature extraction methods before entering the classification phase. Results Images of the extracted features are obtained as shown in (Appendix A) , where the tables show taking a window from the image with a size of 8 * 8 and placing the pixel values after performing feature extraction methods on it. Therefore, these results will be discussed as follows, according to the feature extraction method that used:

4.3.1.1 Results without using Feature Extraction Method

In this style of implementation, the dataset was entered after applying pre-processing to the classification stage directly without applying any method of feature extraction. Figure (4.1) displays the dataset before entering the classification stage.

4.3.1.2 Results of Feature Extraction by HOG Method

The HOG descriptor method was used by the proposed system as a feature extractor applying the Sobel operator. The images resulting from this method show how the gradient directions are distributed in local parts of the image. The result of applying the HOG descriptor to the image is a feature vector that represents the local object's appearance and shape within the analyzed region. It gave good results in detecting the object and determining its behavior. The results of this method appeared as shown in Figure (4.2).



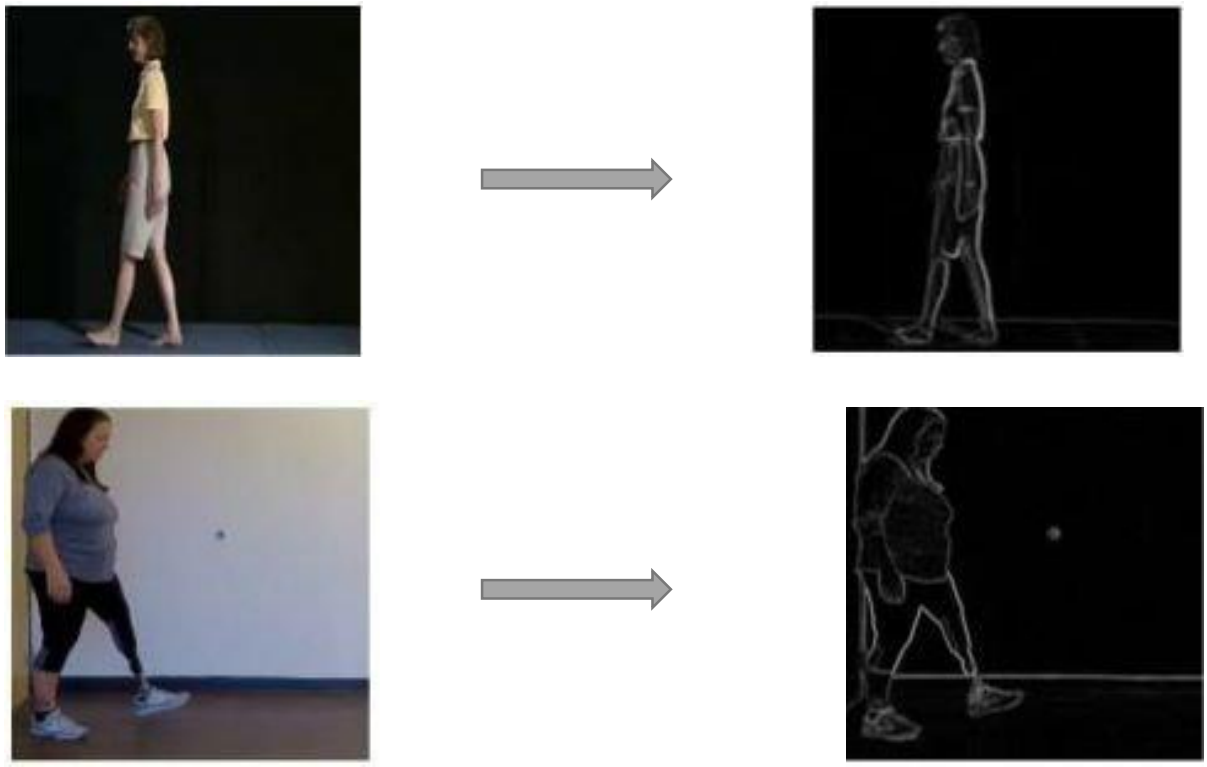


Figure (4.2): Results of Feature Extraction by HOG Method

4.3.1.3 Results of Feature Extraction by LBP Method

When applying the LBP descriptor method to obtain the features extracted from the dataset entered into it, important results were reached, the effect of which later appeared on the accuracy of classification, and it is better than the previous two methods, as the range of the histogram using the LBP method is in the range from (0 to 255). Figure (4.3) show the image histogram when using the LBP descriptor.

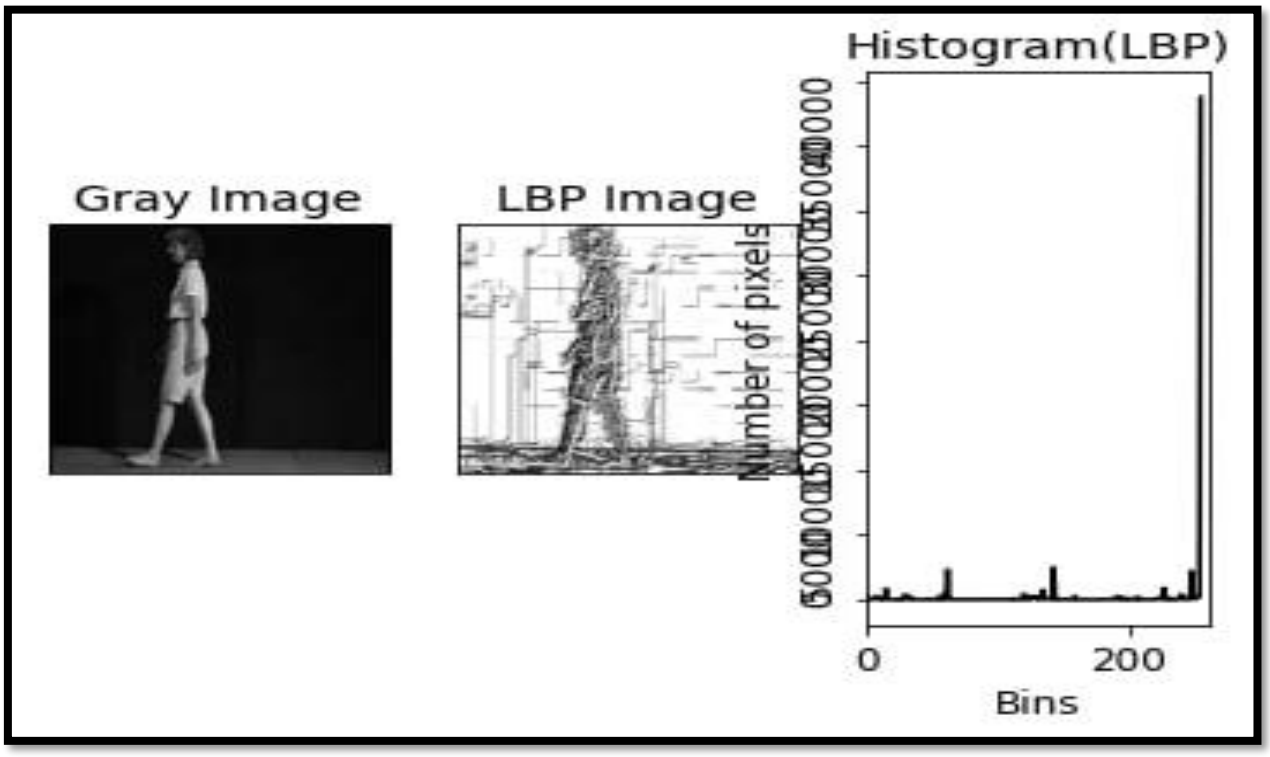


Figure (4.3): Image histogram by using LBP Method

The resulting LBP histogram captures the local texture patterns in the image. The images obtained in this way were clearer and had a clear impact on the results obtained with the accuracy of classification, which made them distinct from the previous two methods. In Figure (4.4) are show the results of extracting features from the dataset when using the LBP descriptor.

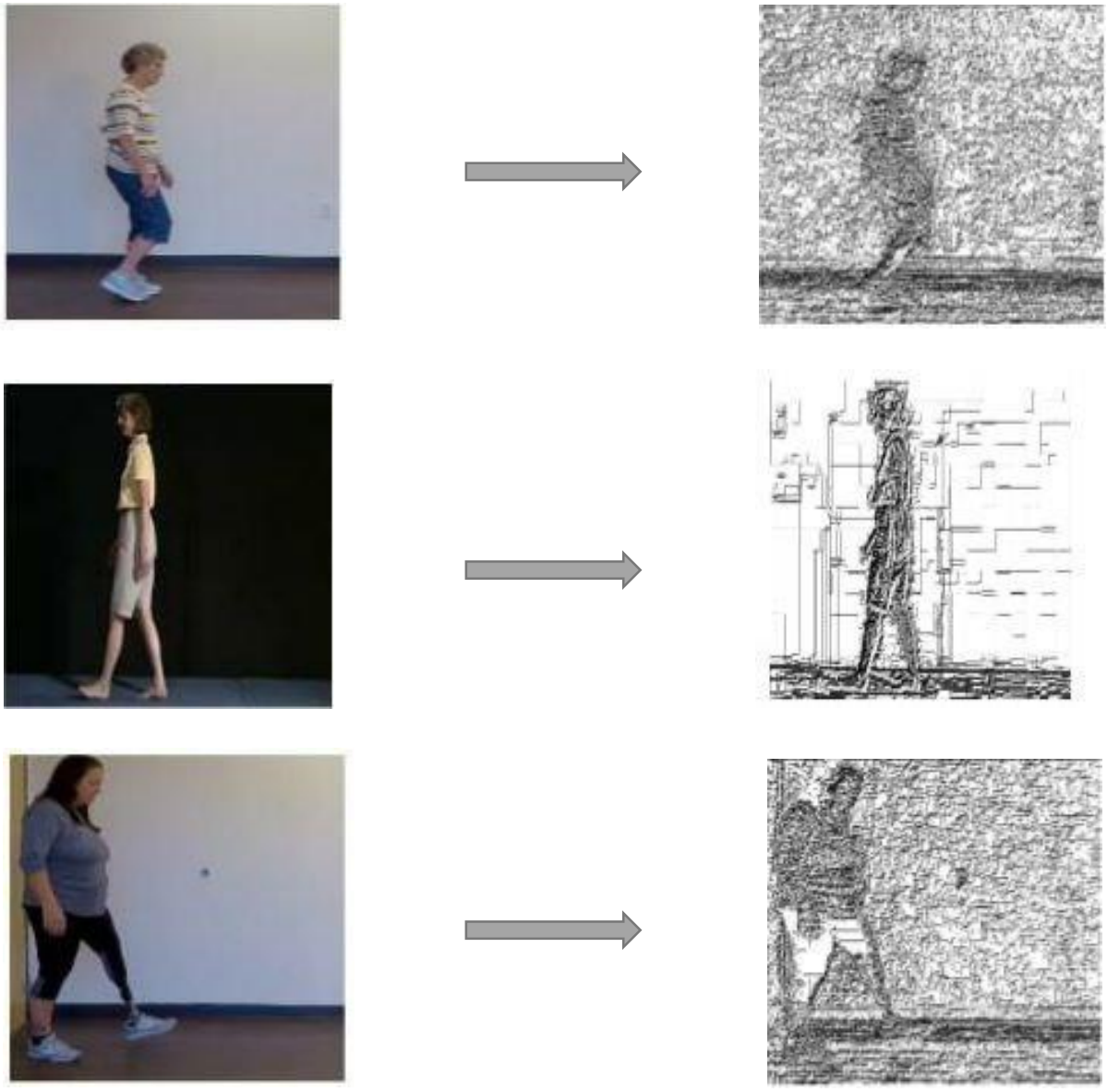


Figure (4.4): Results of Feature Extraction by LBP Method

4.3.1.4 Results of Feature Extraction by CS-LBP Method

When applying the CS-LBP descriptor method to obtain the features extracted from the dataset entered in it, important results were obtained, which will later affect the classification accuracy and the results were better than all previous methods, where the range of the histogram using the CS-LBP method is located in the range

from (0 to 15). In Figure (4.5), the image histogram is shown when using the CS-LBP descriptor.

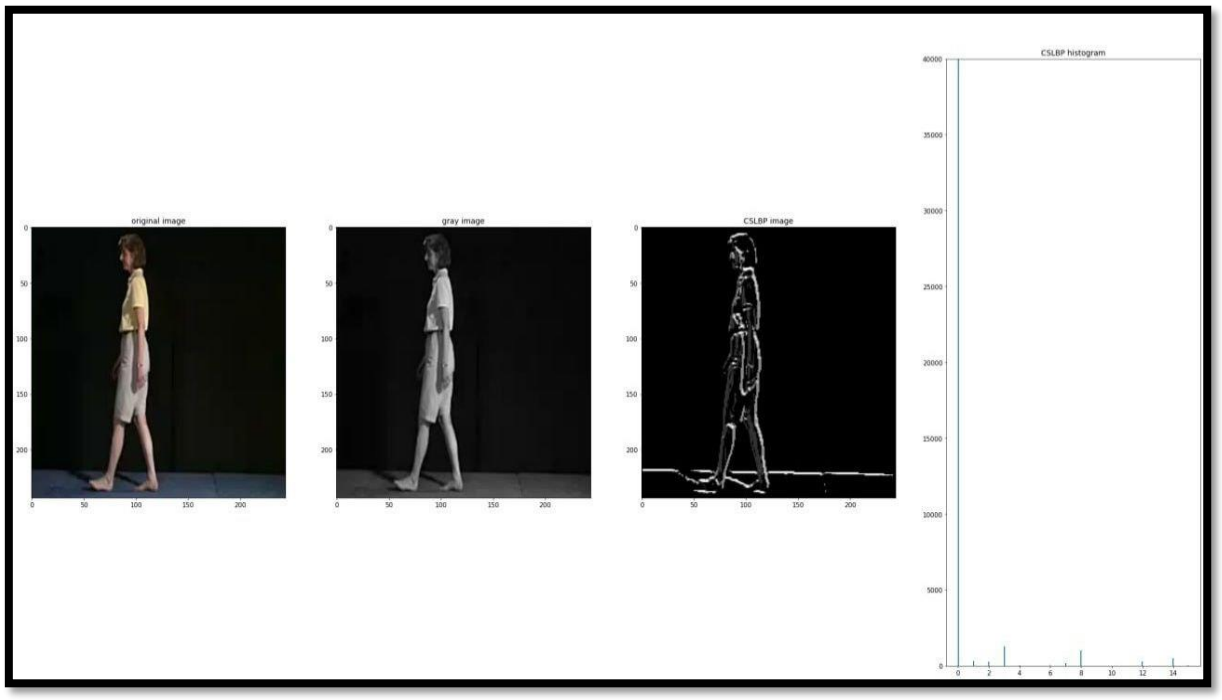


Figure (4.5): Image histogram by using CS-LBP Method

The resulting images when applying this method represent a histogram representing the distribution of different CS-LBP patterns in the image. Being an improved method for LBP that addresses some of its limitations, it outperforms standard LBP in difficult scenarios such as noisy environments, fluctuating lighting, and low-quality images due to enhanced rotational stability, partial data tolerance, and lower noise exposure. This is done by comparing pixel density values. In a local neighborhood around the central pixel, this appeared in the results obtained and the clarity of the extracted features, which in turn will positively affect the classification accuracy, which led to the superiority of this method and giving impressive results that distinguished it from previous methods. In Figure (4.6) are show the results of extracting features from the dataset when using the CS-LBP descriptor.

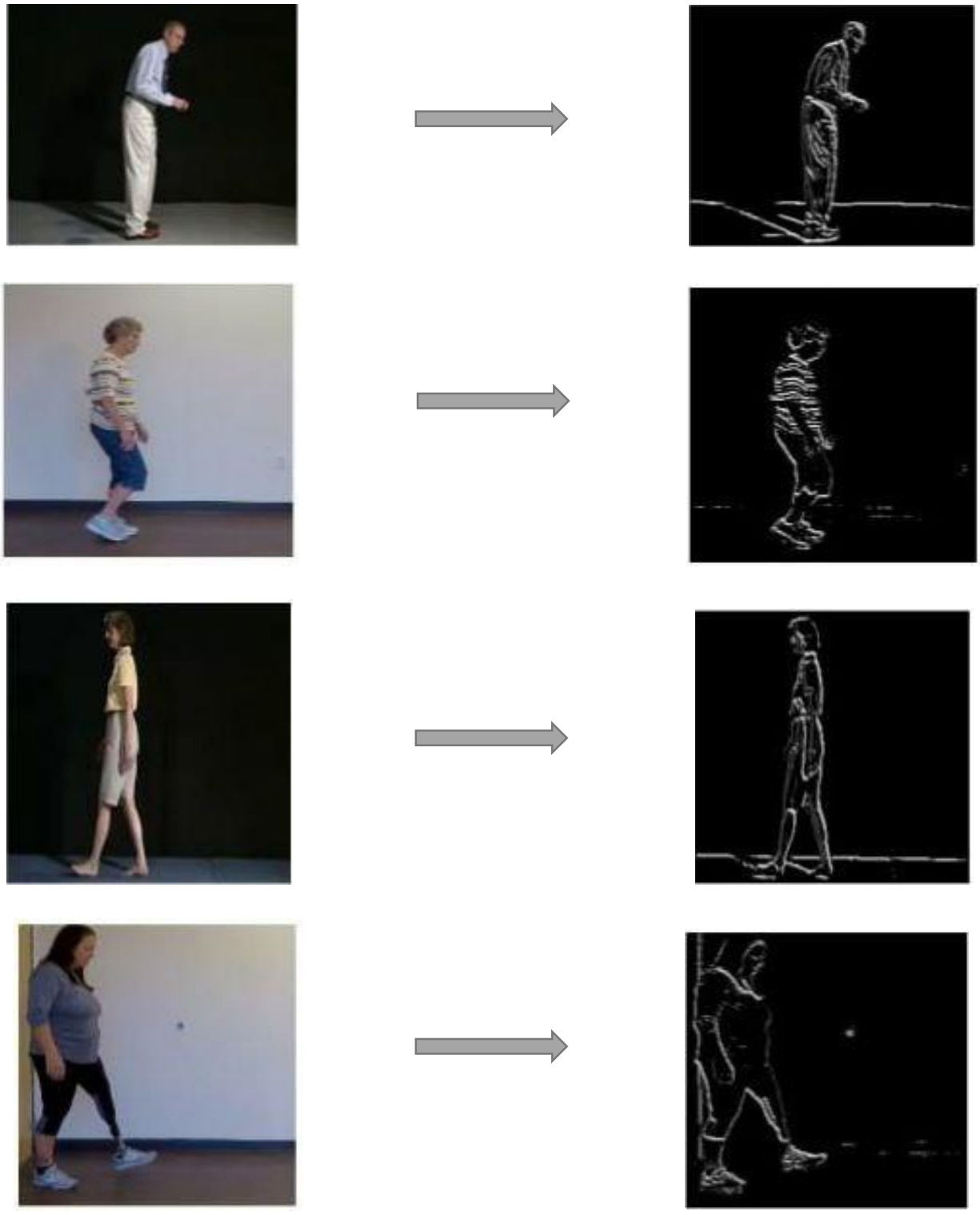


Figure (4.6): Results of Feature Extraction by CS-LBP Method

4.3.1.5 Results of Feature Extraction by XCS-LBP Method

The XCS-LBP descriptor method was used by the system to extract features from the dataset entered into it. This method was found to be more resilient to noise and light differences because it combines the best aspects of the original LBP and related CS-LBP techniques. It also provides a short histogram with a range from (0 to 15). Results were obtained that will later affect the classification accuracy. The results were better than the previous methods. In Figure (4.7) the image histogram is shown when the XCS-LBP descriptor is used.

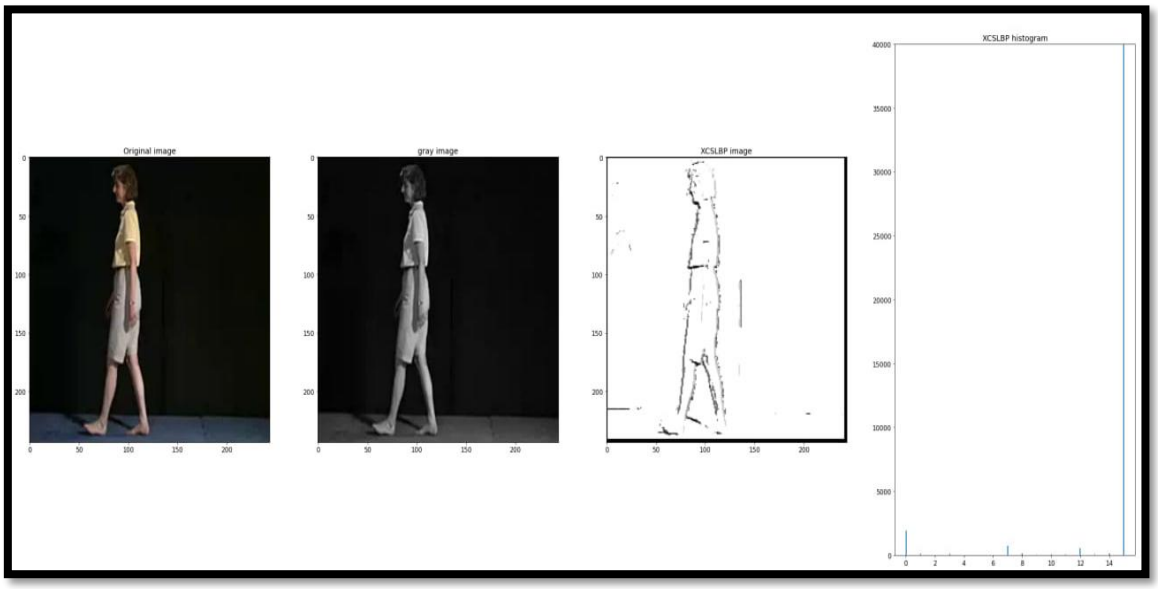


Figure (4.7): Image histogram by using XCS-LBP Method

The resulting images when applying this method represent a graph representing the distribution of the different XCS-LBP patterns in the image. The images resulting from this method are similar to the results of the previous method, as images were obtained that showed the clarity of the extracted features, which in turn will positively affect the classification accuracy, which also led to superior This method gives amazing results that distinguish it from previous methods. Figure (4.8) below

shows the results of extracting features from the dataset when using the XCS-LBP descriptor.

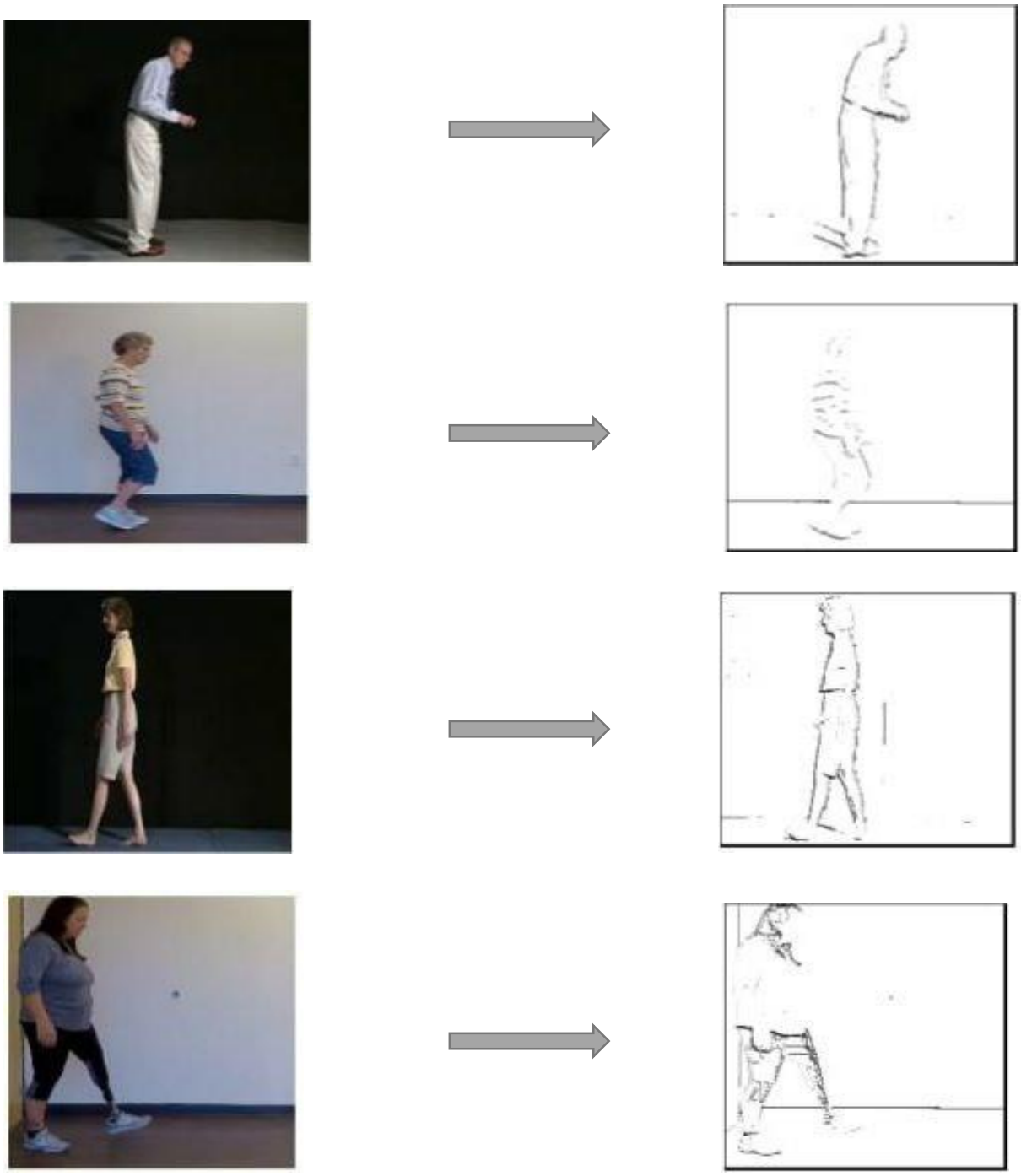


Figure (4.8): Results of Feature Extraction by XCS-LBP Method

4.3.2 Results of Classification Stage

The CNN classifier was used to classify normal and abnormal based on human gait patterns. In the training phase, images classification uses the Adam optimization and reduce the learning rate optimizes the validation loss value when model performance stops increase (decrease on plateau), batch size = 64, and stop early to find the best number of epochs (maximum=20). The results of this stage were divided into five styles according to the input images and the method of extracting features. Therefore, the explanation of each stage by showing the results of training and testing will be as follows:

4.3.2.1 Results of Classification Without Feature Extraction Method

The results of this method are obtained after performing pre-processing on the dataset and only preprocessing stages were performed without feature extraction. The total number of images was (4909), some of which were normal and others were abnormal. After that, the dataset was divided into 20% for testing and 80% for training. It stabilized at a certain value after 10 epochs and then training was stopped. This form took approximately 52,304 seconds to train. Table (4.1) displays values the accuracy and loss learning for the train and the validation samples.

Table (4.1): shows values the loss learning and accuracy learning of train and validation sample (without feature extraction method).

Epoch	Loss	Val Loss	Acc	Val Acc
1	2.1739	2.1539	0.5511	0.6517
2	1.4327	1.7245	0.5411	0.6517
3	1.0855	0.7325	0.5574	0.6517

Epoch	Loss	Val Loss	Acc	Val Acc
4	1.2539	0.6681	0.5465	0.6517
5	0.7796	0.7432	0.5750	0.6517
6	0.7397	0.8549	0.5814	0.6517
7	0.7103	0.6465	0.6017	0.6517
8	0.6952	0.6608	0.6033	0.6517
9	0.6798	0.6468	0.6208	0.6517
10	0.6637	0.6513	0.6404	0.6517

In the testing phase, the classification accuracy result was (65%), and the table (4.2) shows the results of evaluation metrics.

Table (4.2): Results obtained when the without feature extraction method

Precision	Recall	F1-score	Accuracy
0.33	**0.50**	**0.39**	**0.65**

Figure (4.9) represent the confusion matrix:

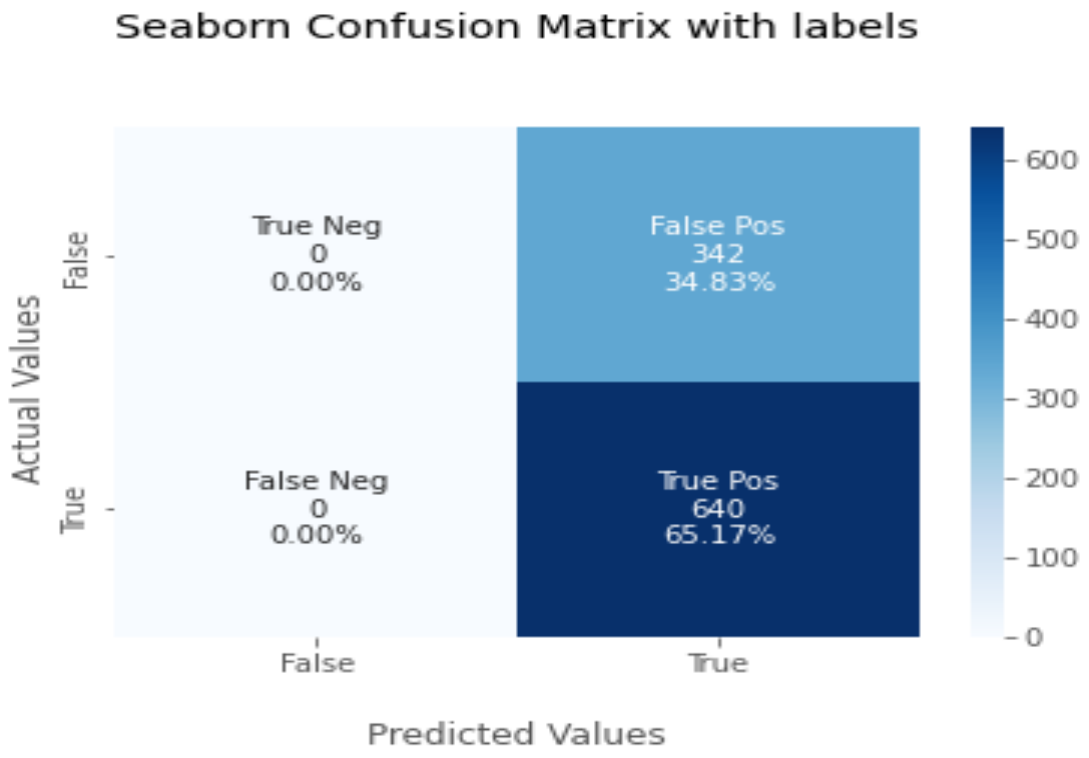


Figure (4.9) The test sample's confusion matrix

4.3.2.2 Results of Classification by HOG Feature Extraction Method

The results of this method are obtained after performing pre-processing on the data set and then features are extracted from the data using the HOG method, noting that the number available is (4909) normal and abnormal images. The data was analyzed, divided into 20% for test data and 80% of the training data, stabilizes at a certain value after 20 epochs and then stops training. The method took approximately (7843 seconds) to train. The learning values for accuracy and loss for training and validation samples are shown in Table (4.3).

Table (4.3): shows values the loss learning and accuracy learning of train and validation sample (using HOG method).

Epoch	Loss	Val Loss	Acc	Val Acc
1	1.4355	0.5115	0.6883	0.8269
2	0.4666	0.3602	0.8034	0.8411
3	0.5834	0.3230	0.7884	0.8798
4	0.5208	0.4013	0.8128	0.8452
5	0.3808	0.3353	0.8582	0.8625
6	0.3857	0.2650	0.8510	0.9022
7	0.3394	0.3056	0.8696	0.8859
8	0.3403	0.2520	0.8671	0.9165
9	0.2870	0.2961	0.8997	0.8941
10	0.3378	0.3185	0.8711	0.8849
11	0.3063	0.2423	0.8887	0.9145
12	0.3360	0.2377	0.8719	0.9226
13	0.2898	0.2870	0.8966	0.8971
14	0.3024	0.3714	0.8831	0.8432
15	0.2927	0.2786	0.8938	0.9022
16	0.2838	0.2610	0.8933	0.9043
17	0.2974	0.4292	0.8869	0.7709
18	0.2815	0.2446	0.8951	0.9196
19	0.2901	0.2218	0.8979	0.9308
20	0.2746	0.2515	0.9055	0.9084

In the testing phase, the classification accuracy result for this phase was (91%), and the table (4.4) shows the results of evaluation metrics.

Table (4.4): Results obtained when using HOG method

Precision	Recall	F1-score	Accuracy
0.93	**0.87**	**0.89**	**0.91**

Figure (4.10) represent the confusion matrix:

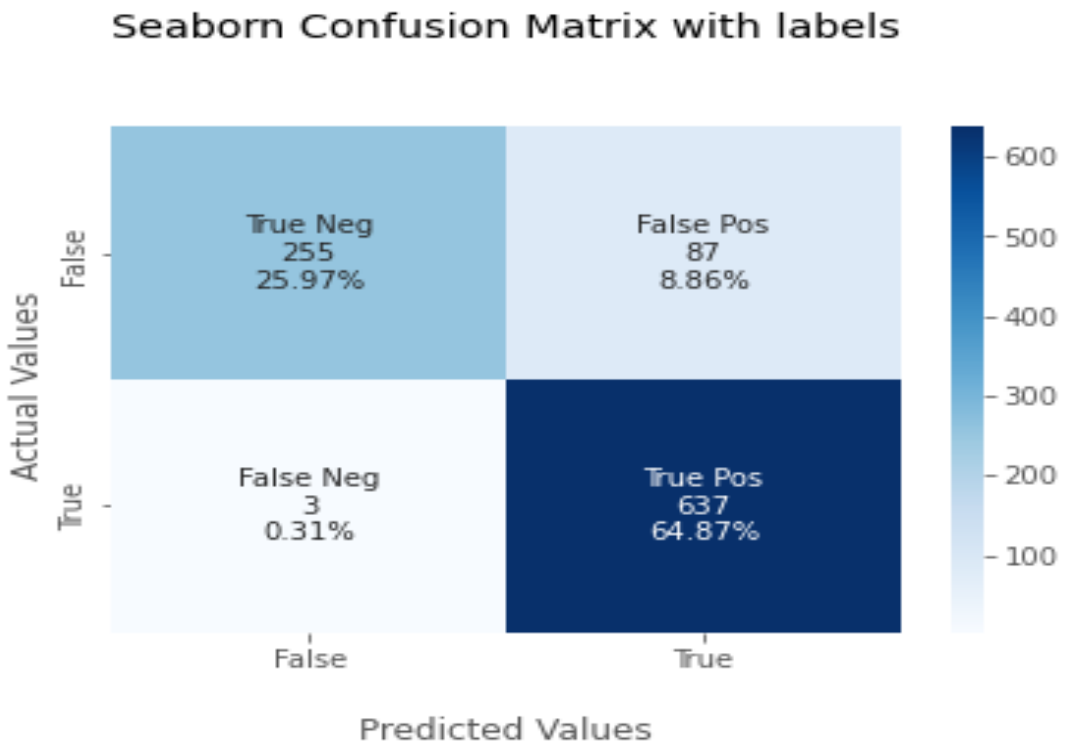


Figure (4.10) the test sample's confusion matrix

4.3.2.3 Results of Classification by LBP Feature Extraction Method

The results of this method are obtained after performing pre-processing on the data set and then features are extracted from the data using the LBP method. The available images are (4909) normal and abnormal images, and the dataset is divided

into 20% for the testing and 80% for the training. It stabilizes at a certain value after 5 periods and then stops training. The method took approximately 2105 seconds to train. The learning values for accuracy and loss for training and validation samples are shown in Table (4.5).

Table (4.5): shows values the loss learning and accuracy learning of train and validation sample (using LBP method).

Epoch	Loss	Val Loss	Acc	Val Acc
1	1.5307	0.3220	0.7054	0.8778
2	0.3500	0.2212	0.8528	0.9063
3	0.3021	0.2101	0.8775	0.9134
4	0.3519	0.7566	0.8752	0.6945
5	0.3356	0.1594	0.8857	0.9369

In the testing phase, the classification accuracy result for this phase was (94%), and the table (4.6) shows the results of evaluation metrics.

Table (4.6): Results obtained when using LBP method

Precision	Recall	F1-score	Accuracy
0.94	**0.92**	**0.93**	**0.94**

Figure (4.11) represent the confusion matrix:

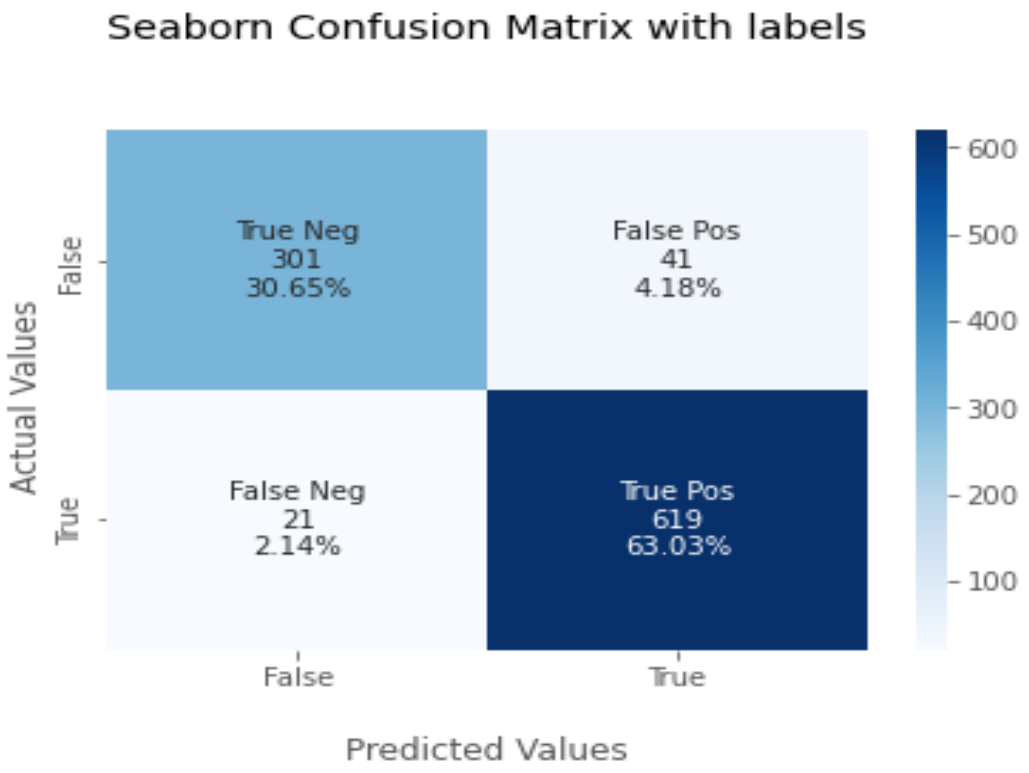


Figure (4.11) the test sample's confusion matrix

4.3.2.4 Results of Classification by CS-LBP Feature Extraction Method

The results of this method are obtained after performing pre-processing on the data set and then features are extracted from the data using the CS-LBP method, and we had (4909) normal and abnormal images, then the data was divided into 20% for test data and 80% for training data, it stabilizes at a certain value after 5 epochs and then stops training. Where in the method it took approximately (1953 seconds) for training. The learning values for accuracy and loss of train and validation samples are shown in Table (4.7).

Table (4.7): shows values the loss learning and accuracy learning of train and validation sample (using CS-LBP method).

Epoch	Loss	Val Loss	Acc	Val Acc
1	1.1824	0.1527	0.9689	0.9929
2	0.0616	0.058	0.9964	0.9959
3	0.0039	0.0464	0.999	0.998
4	0.0341	0.302	0.9982	0.9939
5	0.0625	0.419	0.9969	0.9929

In the testing phase, the result of classification accuracy for this phase was (99%), and it was noted through the results that the best percentage of accuracy in detecting abnormal behavior was achieved from the previous methods in this case. Table (4.8) shows the results of the evaluation measures.

Table (4.8): Results obtained when using CS-LBP method

Precision	Recall	F1-score	Accuracy
0.99	**0.99**	**0.99**	**0.99**

Figure (4.12) represent the confusion matrix:

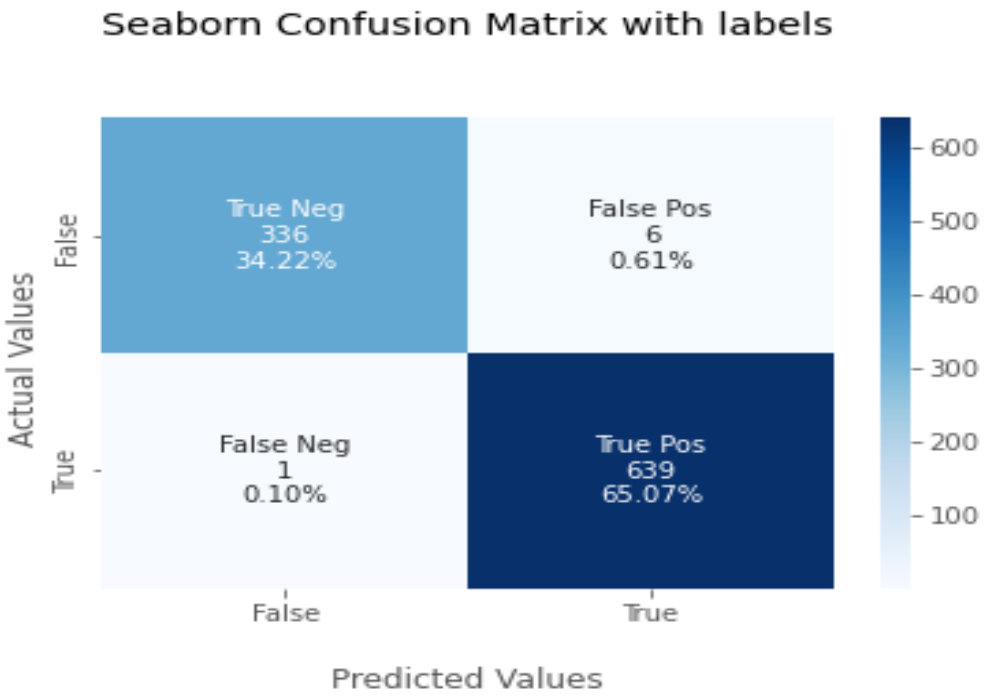


Figure (4.12) the test sample's confusion matrix

4.3.2.5 Results of Classification by XCS-LBP Feature Extraction Method

The results of this method are obtained after performing pre-processing on the data set and then features are extracted from the data using the XCS-LBP method, and we had (4909) normal and abnormal images, then the data was divided into 20% for test data and 80% for training data, it stabilizes at a certain value after 5 epochs and then stops training. This method took approximately (1943 seconds) for training. The learning values for accuracy and loss of train and validation samples are shown in Table (4.9).

Table (4.9): shows values the loss learning and accuracy learning of train and validation sample (using XCS-LBP method).

Epoch	Loss	Val Loss	Acc	Val Acc
1	1.3803	0.0596	0.9638	0.9939
2	0.0321	0.0167	0.9975	0.9980
3	0.0125	0.0569	0.9990	0.9959
4	0.0597	0.0481	0.9964	0.9949
5	0.0088	0.0885	0.9992	0.9929

In the testing phase, the classification accuracy result for this phase was (99%), and it was noted through the results that it was shown to be the best accuracy rate in detecting abnormal behavior if compared to the previous methods in this case, and the results were similar to the CS-LBP method. Table (4.10) shows the results of the evaluation measures.

Table (4.10): Results obtained when using XCS-LBP method

Precision	Recall	F1-score	Accuracy
0.99	**0.99**	**0.99**	**0.99**

Figure (4.13) represent the confusion matrix:

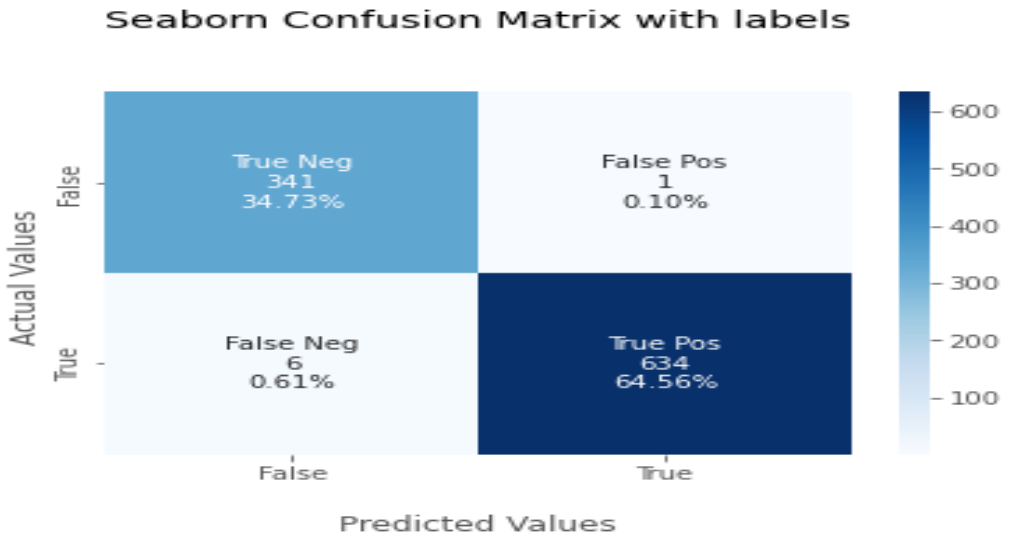
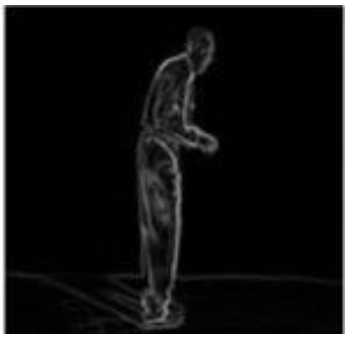


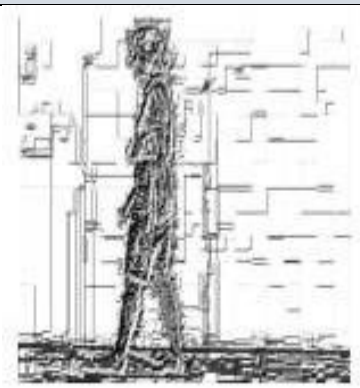
Figure (4.13) the test sample's confusion matrix

Through the previous results, a comparison between the methods used can be summarized, as shown in the Table (4.11).

Table (4.11): Summary comparison between the methods that used in the proposed system

No.	Method	Feature Extraction	Accuracy	Time of Training	Time of Testing
1	Without Feature Extraction	 Normal	65%	52304 sec	1059 sec

No.	Method	Feature Extraction	Accuracy	Time of Training	Time of Testing
		 **Abnormal**			
2	**Histogram of Oriented Gradients (HOG)**	 **Normal**  **Abnormal**	**91%**	**7843 sec**	**76 sec**

No.	Method	Feature Extraction	Accuracy	Time of Training	Time of Testing
3	Local Binary Pattern (LBP)	 Normal  Abnormal	94%	2105 sec	75 sec
4	Center-Symmetric Local Binary Pattern (CS-LBP)	 Normal	99%	1953 sec	76 sec

No.	Method	Feature Extraction	Accuracy	Time of Training	Time of Testing
		 **Abnormal**			
5	**Extended Center-Symmetric Local Binary Pattern (XCS-LBP)**	 **Normal**  **Abnormal**	**99%**	**1943 sec**	**78 sec**

4.4 Comparison with other Published Methods

In this section, a comparison is made with other published works. The proposed system represents a major advance in the field of detecting abnormal behavior based on walking using CNNs. Through the use of features extraction techniques from images and these techniques made the work more accurate, less time consuming, achieving satisfactory results and outperforming previous studies in this field, our work contributes valuable insights and methodologies to improve the accuracy and reliability of abnormal behavior detection systems. Table (4.12) shows a comparison of the Results of the proposed work with other published works.

Table (4.12): Comparison of our work with other published methods

Reference	Research Title	Technique Used	Dataset	Accuracy
P. Nithyakani et al., in 2019 [10]	Human Gait Recognition using Deep Convolutional Neural Network	Deep CNN	TUM Gaid	94.7 %
J. Gao et al., in 2019 [11]	Abnormal Gait Recognition Algorithm Based on LSTM-CNN Fusion Network	LSTM- CNN	manually collected from the inertia sensors	93.1 %
A. R. Hawas et al., in 2019 [12]	Gait identification by convolutional neural networks and optical flow	CNN	CASIA B Gait	95 %

Reference	Research Title	Technique Used	Dataset	Accuracy
F. Saleem et al., in 2021 [14]	Human gait recognition: A single stream optimal deep learning features fusion	CNN	CASIA B Gait	89 %
M. A. Khan et al., in 2021 [15]	Human gait analysis for osteoarthritis prediction: a framework of deep learning and kernel extreme learning machine	CNN	CASIA B Gait	96 %
P. Chophuk et al. in 2023 [18]	Theft Detection by Patterns of Walking Behavior using Motion-based Artificial Intelligence	Bi-LSTM	UCF-Crime	97 %
Our proposed method	**Abnormal Gait Patterns Classification Based on Spatial Image Features and Deep Learning Technique**	**CNN+VGG16**	**Real dataset**	**99 %**

Chapter Five

Conclusions and Suggestions for Future Work

Chapter Five

Conclusions and Suggestions for Future Work

5.1 Conclusions

The main conclusions from the results obtained from the use of the proposed system for Abnormal Gait Patterns Classification Based on Spatial Image Features and Deep Learning Technique, a number of conclusions can be drawn based on the experiments conducted:

1. Pre-possessing is considered an important stage within the stages of the proposed work, which ensures that the images are suitable for the upcoming stages, as it is the first and most important step in improving the results of the work of the feature extraction stage and thus improving the results of image classification.
2. In the proposed system, local spatial feature extraction methods were used to extract the feature, which is an important step in obtaining important and relevant data that was used in the detection stage. These methods are (HOG, LBP, CS-LBP, XCS-LBP), and satisfactory results were obtained in terms of classification accuracy and training time on the data using the CS-LBP and XCS-LBP methods. The accuracy of both methods was 99%.
3. In the proposed system and through the experimental results, it is clear that the CNN-Vgg16 is effective in obtaining better results in abnormal gait patterns classification based on gait analysis.
4. The proposed method is feasible and can be applied in early classify of health. Some diseases may be detected in their early stages by observing a person's steps so that patients can get timely treatment, and a lot of related areas to spot abnormal gait patterns based on human gait.

5.2 Suggestions for Future Work

It is possible to suggest a number of ideas in order to implement them in the future to improve the effectiveness of the proposed work and develop its performance.

1. It can be suggested to add multiple categories of target class to gait patterns, for example, classifying medical images of gait according to the type of disease.
2. Try to discover the person's gait patterns based on the type of circumstance practices or lives in, for example when he carries a bag or wears a hat and so on.
3. Trying to classify and distinguish gait patterns of more than one person at the same time.
4. Increase the capacity of the proposed system by taking advantage of a huge dataset of images.
5. Use alternative frequency domain feature extraction techniques, such as feature sets extracted from wavelet transforms, to develop the proposed system for the application.

References

References

[1] H. K. Jameel and B. N. Dhannoon, “Gait Recognition Based on Deep Learning,” *Iraqi J. Sci.*, vol. 63, no. 1, pp. 397–408, Jan. 2022, doi: 10.24996/ijs.2022.63.1.36.

[2] I. Bouchrika, M. Goffredo, J. Carter, and M. Nixon, “On using gait in forensic biometrics,” *Journal of Forensic Sciences*, vol. 56, no. 4. pp. 882–889, 2011. doi: 10.1111/j.1556-4029.2011.01793.x.

[3] C. Wang, X. Wu, N. Li, and Y. L. Chen, “Abnormal detection based on gait analysis,” *Proc. World Congr. Intell. Control Autom.*, no. 61005012, pp. 4859–4864, 2012, doi: 10.1109/WCICA.2012.6359398.

[4] G. Batchuluun, Y. G. Kim, J. H. Kim, H. G. Hong, and K. R. Park, “Robust behavior recognition in intelligent surveillance environments,” *Sensors (Switzerland)*, vol. 16, no. 7, 2016, doi: 10.3390/s16071010.

[5] K. Muhammad, S. Khan, M. Elhoseny, S. Hassan Ahmed, and S. Wook Baik, “Efficient Fire Detection for Uncertain Surveillance Environment,” *IEEE Trans. Ind. Informatics*, vol. 15, no. 5, pp. 3113–3122, 2019, doi: 10.1109/TII.2019.2897594.

[6] F. U. M. Ullah, A. Ullah, K. Muhammad, I. U. Haq, and S. W. Baik, “Violence detection using spatiotemporal features with 3D convolutional neural network,” *Sensors*, vol. 19, no. 11, pp. 1–15, 2019, doi: 10.3390/s19112472.

[7] K. K. Verma, B. M. Singh, A. Dixit, “A Review Of Supervised And Unsupervised Machine Learning Techniques For Suspicious Behavior Recognition In Intelligent Surveillance System” Int. J. Inf. Technol., vol. 14, no. 1, pp. 397–410, 2022, doi: 10.1007/s41870-019-00364-0.

[8] E. Bermejo Nievas, O. Deniz Suarez, G. Bueno García, and R. Sukthankar, "Violence detection in video using computer vision techniques," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6855 LNCS, no. PART 2, pp. 332–339, 2011, doi: 10.1007/978-3-642-23678-5_39.

[9] K. Shiraga, Y. Makihara, D. Muramatsu, T. Echigo, and Y. Yagi, "GEINet: View-invariant gait recognition using a convolutional neural network," *2016 Int. Conf. Biometrics, ICB 2016*, 2016, doi: 10.1109/ICB.2016.7550060.

[10] P. Nithyakani, A. Shanthini, and G. Ponsam, "Human Gait Recognition using Deep Convolutional Neural Network," *2019 Proc. 3rd Int. Conf. Comput. Commun. Technol. ICCCT 2019*, pp. 208–211, 2019, doi: 10.1109/ICCCT2.2019.8824836.

[11] J. Gao, P. Gu, Q. Ren, J. Zhang, and X. Song, "Abnormal Gait Recognition Algorithm Based on LSTM-CNN Fusion Network," *IEEE Access*, vol. 7, pp. 163180–163190, 2019, doi: 10.1109/ACCESS.2019.2950254.

[12] A. R. Hawas, H. A. El-Khobby, M. Abd-Elnaby, and F. E. Abd El-Samie, "Gait identification by convolutional neural networks and optical flow," *Multimed. Tools Appl.*, vol. 78, no. 18, pp. 25873–25888, 2019, doi: 10.1007/s11042-019-7638-9.

[13] A. Mehmood, "Abnormal behavior detection in uncrowded videos with two-stream 3d convolutional neural networks," *Appl. Sci.*, vol. 11, no. 8, 2021, doi: 10.3390/app11083523.

[14] F. Saleem *et al.*, "Human gait recognition: A single stream optimal deep learning features fusion," *Sensors*, vol. 21, no. 22, 2021, doi: 10.3390/s21227584.

[15] M. A. Khan *et al.*, “Human gait analysis for osteoarthritis prediction: a framework of deep learning and kernel extreme learning machine,” *Complex Intell. Syst.*, 2021, doi: 10.1007/s40747-020-00244-2.

[16] P. Kuppusamy and V. C. Bharathi, “Human abnormal behavior detection using CNNs in crowded and uncrowded surveillance – A survey,” *Meas. Sensors*, vol. 24, no. July, 2022, doi: 10.1016/j.measen.2022.100510.

[17] C. Filipi Gonçalves Dos Santos *et al.*, “Gait Recognition Based on Deep Learning: A Survey,” *ACM Comput. Surv.*, vol. 55, no. 2, 2023, doi: 10.1145/3490235.

[18] P. Chophuk, P. Boonmee, S. Jiarasuwan, S. Jearasuwan, and P. Bookprakong, “Theft detection by patterns of walking behavior using motion-based artificial intelligence,” no. March, p. 105, 2023, doi: 10.1117/12.2671245.

[19] C. Li, Z. Han, Q. Ye, and J. Jiao, “Visual abnormal behavior detection based on trajectory sparse reconstruction analysis,” *Neurocomputing*, vol. 119, pp. 94–100, 2013, doi: 10.1016/j.neucom.2012.03.040.

[20] Y. Hu, “Design and Implementation of Abnormal Behavior Detection Based on Deep Intelligent Analysis Algorithms in Massive Video Surveillance,” *J. Grid Comput.*, vol. 18, no. 2, pp. 227–237, 2020, doi: 10.1007/s10723-020-09506-2.

[21] L. Devroye and G. L. Wise, “Detection of Abnormal Behavior Via Nonparametric Estimation of the Support,” *SIAM J. Appl. Math.*, vol. 38, no. 3, pp. 480–488, 1980, doi: 10.1137/0138038.

[22] J. Zhang, C. Wu, Y. Wang, and P. Wang, “Detection of abnormal behavior in narrow scene with perspective distortion,” *Mach. Vis. Appl.*, vol. 30, no. 5, pp. 987–998, 2019, doi: 10.1007/s00138-018-0970-7.

[23] C. Wu and Z. Cheng, “A novel detection framework for detecting abnormal human behavior,” *Math. Probl. Eng.*, vol. 2020, 2020, doi: 10.1155/2020/6625695.

[24] M. I. Mokhlespour Esfahani and M. A. Nussbaum, “Using smart garments to differentiate among normal and simulated abnormal gaits,” *J. Biomech.*, vol. 93, pp. 70–76, 2019, doi: 10.1016/j.jbiomech.2019.06.009.

[25] A. Abdelwhab and S. Viriri, “A Survey on Soft Biometrics for Human Identification,” *Mach. Learn. Biometrics*, 2018, doi: 10.5772/intechopen.76021.

[26] Emilio Mordini and Dimitros Tzovaras, *Second Generation Biometrics: The Ethical, Legal and Social Context*. 2012. doi: 10.1007/978-94-007-3892-8.

[27] S. Zulkarnain, S. Idrus, E. Cherrier, C. Rosenberger, and J.-J. Schwartzmann, “A Review on Authentication Methods,” *Aust. J. Basic Appl. Sci.*, vol. 7, no. 5, pp. 95–107, 2013, [Online]. Available: https://hal.archives-ouvertes.fr/hal-00912435%0Ahttps://hal.archives-ouvertes.fr/hal-00912435/document

[28] M. Sharif, M. Raza, J. H. Shah, M. Yasmin, and S. L. Fernandes, “An overview of biometrics methods,” *Handb. Multimed. Inf. Secur. Tech. Appl.*, pp. 15–35, 2019, doi: 10.1007/978-3-030-15887-3_2.

[29] M. H. Khan, F. Li, M. S. Farid, and M. Grzegorzek, “Gait recognition using motion trajectory analysis,” *Adv. Intell. Syst. Comput.*, vol. 578, no. May, pp. 73–82, 2018, doi: 10.1007/978-3-319-59162-9_8.

[30] S. Yu, H. Chen, Q. Wang, L. Shen, and Y. Huang, “Invariant feature extraction for gait recognition using only one uniform model,” *Neurocomputing*, vol. 239, pp. 81–93, 2017, doi: 10.1016/j.neucom.2017.02.006.

[31] T. T. Sruti Das Choudhury, “warwick.ac.uk/lib-publications,” pp. 1–7, 2016.

[32] W. K. Mutlag, S. K. Ali, Z. M. Aydam, and B. H. Taher, “Feature Extraction Methods: A Review,” *J. Phys. Conf. Ser.*, vol. 1591, no. 1, 2020, doi: 10.1088/1742-6596/1591/1/012028.

[33] N. Baaziz, O. Abahmane, and R. Missaoui, “Texture feature extraction in the spatial-frequency domain for content-based image retrieval,” pp. 1–19, 2010, [Online]. Available: http://arxiv.org/abs/1012.5208

[34] T. Neubert, C. Kraetzer, and J. Dittmann, “A face morphing detection concept with a frequency and a spatial domain feature space for images on eMRTD,” *IH MMSec 2019 - Proc. ACM Work. Inf. Hiding Multimed. Secur.*, pp. 95–100, 2019, doi: 10.1145/3335203.3335721.

[35] L. Debiasi, U. Scherhag, C. Rathgeb, A. Uhl, and C. Busch, “PRNU-based detection of morphed face images,” *IWBF 2018 - Proc. 2018 6th Int. Work. Biometrics Forensics*, pp. 1–7, 2018, doi: 10.1109/IWBF.2018.8401555.

[36] D. Zhang, Q. Guo, G. Wu, and D. Shen, “Sparse patch-based label fusion for multi-atlas segmentation,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7509 LNCS, pp. 94–102, 2012, doi: 10.1007/978-3-642-33530-3_8.

[37] W. Zhou, S. Gao, L. Zhang, and X. Lou, “Histogram of Oriented Gradients Feature Extraction from Raw Bayer Pattern Images,” *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 67, no. 5, pp. 946–950, 2020, doi: 10.1109/TCSII.2020.2980557.

[38] T. Ojala, M. Pietikäinen, and D. Harwood, “A comparative study of texture measures with classification based on feature distributions,” *Pattern Recognit.*, vol. 29, no. 1, pp. 51–59, 1996, doi: 10.1016/0031-3203(95)00067-4.

[39] B. Jun, T. Kim, and D. Kim, “A compact local binary pattern using

maximization of mutual information for face analysis," *Pattern Recognit.*, vol. 44, no. 3, pp. 532–543, 2011, doi: 10.1016/j.patcog.2010.10.008.

[40] S. R. Zhou, J. P. Yin, and J. M. Zhang, "Local binary pattern (LBP) and local phase quantization (LBQ) based on Gabor filter for face representation," *Neurocomputing*, vol. 116, pp. 260–264, 2013, doi: 10.1016/j.neucom.2012.05.036.

[41] B. Yang and S. Chen, "A comparative study on local binary pattern (LBP) based face recognition: LBP histogram versus LBP image," *Neurocomputing*, vol. 120, pp. 365–379, 2013, doi: 10.1016/j.neucom.2012.10.032.

[42] K.Meena and D. A. Suruliandi, "Local Binary Patterns and Its," pp. 782–786, 2011.

[43] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern Recognit.*, vol. 42, no. 3, pp. 425–436, 2009, doi: 10.1016/j.patcog.2008.08.014.

[44] Z. Xiong, M. Liu, and Q. Guo, "Finger Vein Recognition Method Based on Center-Symmetric Local Binary Pattern," *Proc. 2019 IEEE Int. Conf. Artif. Intell. Comput. Appl. ICAICA 2019*, pp. 262–266, 2019, doi: 10.1109/ICAICA.2019.8873475.

[45] C. Silva, T. Bouwmans, and C. Frélicot, "An extended center-symmetric local binary pattern for background modeling and subtraction in videos," *VISAPP 2015 - 10th Int. Conf. Comput. Vis. Theory Appl. VISIGRAPP, Proc.*, vol. 1, pp. 395–402, 2015, doi: 10.5220/0005266303950402.

[46] S. Engineering and S. Lindqvist, "A Machine Learning Approach to Indoor Localization Data Mining," 2020.

[47] M. Batta, "Machine Learning Algorithms - A Review," *Int. J. Sci. Res.*, vol.

18, no. 8, pp. 381–386, 2018, doi: 10.21275/ART20203995.

[48] L. Deng, “An overview of deep-structured learning for information processing,” *APSIPA ASC 2011 - Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. 2011*, pp. 138–151, 2011.

[49] I. G. and Y. B. and A. Courville, *Deep learning: A practitioner's approach*, vol. 29, no. 7553. 2016. [Online]. Available: http://deeplearning.net/

[50] C. Filipi Gonçalves Dos Santos *et al.*, “Network attacks detection methods based on deep learning techniques: A survey,” *Secur. Commun. Networks*, vol. 2020, 2020, doi: 10.1155/2020/8872923.

[51] J. Heidari, “Classifying Material Defects with Convolutional Neural Networks and Image Processing,” *Uppsala Univ.*, p. 56, 2019, [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-387797%0ARefrence66%0A

[52] M. A. Wani, F. A. Bhat, S. Afzal, and A. I. Khan, *Advances in Deep Learning*, vol. 57. 2019. doi: 10.1007/978-981-13-6794-6.

[53] S. J. S. Laith Abualigah, Putra Sumari, “A Novel Deep Learning Pipeline Architecture based on CNN to Detect Covid-19 in Chest X-ray Images,” *Turkish J. Comput. Math. Educ.*, vol. 12, no. 6, pp. 2001–2011, 2021, doi: 10.17762/turcomat.v12i6.4804.

[54] D. Bhatt *et al.*, “Cnn variants for computer vision: History, architecture, application, challenges and future scope,” *Electron.*, vol. 10, no. 20, pp. 1–28, 2021, doi: 10.3390/electronics10202470.

[55] M. Ayi, “Rmnv2: Reduced Mobilenet V2 An Efficient Lightweight Model For Hardware Deployment,” vol. 21, no. 1, pp. 1–9, 2020.

[56] J. Luis and M. Narvaez, “Adaptation of a Deep Learning Algorithm for Traffic

Sign Detection," *J. West. Univ.*, vol. 6, no. September, pp. 1–55, 2019.

[57] R. K. G. D. & K. T. Rikiya Yamashita, Mizuho Nishio, "Convolutional neural networks: an overview and application in radiology https://doi.org/10.1007/s13244-018-0639-9," *Springer*, vol. 195, pp. 21–30, 2018, [Online]. Available: https://www.kaggle.com/datasets/coledie/qr-codes%0A

[58] R. Patil, V. Patil, A. Bahuguna, and G. Datkhile, "Indian Sign Language Recognition using Convolutional Neural Network," *ITM Web Conf.*, vol. 40, no. July, p. 03004, 2021, doi: 10.1051/itmconf/20214003004.

[59] Y. Patankar, M. Mirge, V. Edakhe, P. Gohokar, R. Malpe, and P. Singam, "Real-Time Waste Object Segregation Using Convolutional Neural Network," *Int. J. Comput. Sci. Mob. Comput.*, vol. 11, no. 2, pp. 85–88, 2022, doi: 10.47760/ijcsmc.2022.v11i02.010.

[60] P. Desai, J. Pujari, C. Sujatha, A. Kamble, and A. Kambli, "Hybrid Approach for Content-Based Image Retrieval using VGG16 Layered Architecture and SVM: An Application of Deep Learning," *SN Comput. Sci.*, vol. 2, no. 3, 2021, doi: 10.1007/s42979-021-00529-4.

[61] D. Acharya, W. Yan, and K. Khoshelham, "Real-time image-based parking occupancy detection using deep learning," *CEUR Workshop Proc.*, vol. 2087, no. April, pp. 33–40, 2018.

[62] J. Qi, J. Du, S. M. Siniscalchi, X. Ma, and C. H. Lee, "On Mean Absolute Error for Deep Neural Network Based Vector-to-Vector Regression," *IEEE Signal Process. Lett.*, vol. 27, no. c, pp. 1485–1489, 2020, doi: 10.1109/LSP.2020.3016837.

[63] D. Ramos, J. Franco-Pedroso, A. Lozano-Diez, and J. Gonzalez-Rodriguez,

"Deconstructing cross-entropy for probabilistic binary classifiers," *Entropy*, vol. 20, no. 3, 2018, doi: 10.3390/e20030208.

[64] K. Janocha and W. M. Czarnecki, "On loss functions for deep neural networks in classification," *Schedae Informaticae*, vol. 25, pp. 49–59, 2016, doi: 10.4467/20838476SI.16.004.6185.

[65] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa, "An All-In-One Convolutional Neural Network for Face Analysis," *Proc. - 12th IEEE Int. Conf. Autom. Face Gesture Recognition, FG 2017 - 1st Int. Work. Adapt. Shot Learn. Gesture Underst. Prod. ASL4GUP 2017, Biometrics Wild, Bwild 2017, Heterog. Face Recognition, HFR 2017, Jt. Chall. Domin. Complement. Emot. Recognit. Using Micro Emot. Featur. Head-Pose Estim. DCER HPE 2017 3rd Facial Expr. Recognit. Anal. Challenge, FERA 2017*, pp. 17–24, 2017, doi: 10.1109/FG.2017.137.

[66] J. Schmidhuber, "Deep Learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015, doi: 10.1016/j.neunet.2014.09.003.

[67] Y. Pan *et al.*, "Brain tumor grading based on Neural Networks and Convolutional Neural Networks," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2015-November, pp. 699–702, 2015, doi: 10.1109/EMBC.2015.7318458.

[68] S. Ruder, "An overview of gradient descent optimization algorithms," pp. 1–14, 2016, [Online]. Available: http://arxiv.org/abs/1609.04747

[69] P. Ding, J. Li, L. Wang, M. Wen, and Y. Guan, "HYBRID-CNN: An Efficient Scheme for Abnormal Flow Detection in the SDN-Based Smart Grid," *Secur. Commun. Networks*, vol. 2020, 2020, doi: 10.1155/2020/8850550.

Appendix A

Image elements 8*8 when using Feature Extraction Methods

Figure (A-1): original image

Table (A.1): Image elements 8*8 without using feature extraction methods

0.03529412	0.04313726	0.04313726	0.31004906	0.05098039	0.04313726	0.03210784	0.03137255
0.03529412	0.04313726	0.04313726	0.18431373	0.03921569	0.03921569	0.03137255	0.02745098
0.03529412	0.02352941	0.05931373	0.5867647	0.04509804	0.04313726	0.03921569	0.02745098
0.02745098	0.01960784	0.01862745	0.6014706	0.05490196	0.03921569	0.03921569	0.02745098
0.01568628	0.01960784	0.05882353	0.07720588	0.0627451	0.05490196	0.03921569	0.03137255
0.01568628	0.01960784	0.04142157	0.05588235	0.07058824	0.05490196	0.04313726	0.03529412
0.05122549	0.04607843	0.49558824	0.1392157	0.20808823	0.1644608	0.12794118	0.0254902
0.3389706	0.26666668	0.10343137	0.4389706	0.4019608	0.38333333	0.34436277	0.28259805

Table (A.2): Image elements 8*8 after using HOG method to extract features

0	**0.00098039**	**0**	**0.19128284**	**0.00588235**	**0.00588235**	**0.00177516**	**0**
0.00108191	**0.00294118**	**0.00359237**	**0.09276502**	**0.00098039**	**0**	**0**	**0**
0	**0**	**0.01988548**	**0.52017635**	**0.00784314**	**0**	**0**	**0**
0.00294118	**0**	**0.04010882**	**0.27636293**	**0**	**0**	**0**	**0**
0	**0**	**0.00588236**	**0.02099152**	**0.00392157**	**0**	**0.00294118**	**0**
0	**0**	**0.01670543**	**0.00612745**	**0**	**0**	**0**	**0**
0.05011065	**0.03137255**	**0.25787437**	**0.1598034**	**0.14214696**	**0.13642249**	**0.12322961**	**0.0119238**
0.01059776	**0.07214469**	**0.03730553**	**0.01631693**	**0.01372549**	**0.01960784**	**0.02159465**	**0.01293235**

Table (A.3) Image elements 8*8 after using LBP method to extract features

255	**207**	**255**	**161**	**234**	**234**	**232**	**255**
224	**255**	**91**	**132**	**255**	**255**	**255**	**255**
255	**255**	**225**	**110**	**253**	**255**	**255**	**255**
255	**255**	**237**	**90**	**255**	**255**	**255**	**255**
255	**255**	**255**	**24**	**250**	**255**	**255**	**255**
255	**255**	**247**	**197**	**255**	**255**	**255**	**255**
29	**62**	**15**	**61**	**60**	**60**	**61**	**227**
147	**133**	**63**	**132**	**62**	**62**	**60**	**61**

Table (A.4) Image elements 8*8 after using CS-LBP method to extract features

0	**0**	**0**	**2**	**0**	**0**	**0**	**0**
0	**0**	**0**	**2**	**0**	**0**	**0**	**0**
0	**0**	**0**	**8**	**0**	**0**	**0**	**0**
0	**0**	**0**	**11**	**0**	**0**	**0**	**0**
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	**0**	**5**	**14**	**14**	**14**	**14**	**0**
0	**3**	**0**	**0**	**0**	**0**	**0**	**0**

Table (A.5) Image elements 8*8 after using XCS-LBP method to extract features

15	**15**	**15**	**15**	**15**	**15**	**15**	**15**
15	**15**	**15**	**14**	**15**	**15**	**15**	**15**
15	**15**	**15**	**14**	**15**	**15**	**15**	**15**
15	**15**	**9**	**14**	**15**	**15**	**15**	**15**
15	**15**	**15**	**15**	**15**	**15**	**15**	**15**
15	**15**	**15**	**15**	**15**	**15**	**15**	**15**
15	**15**	**15**	**15**	**15**	**15**	**15**	**15**
15	**15**	**15**	**15**	**15**	**15**	**15**	**15**

Publications

1. Alhussen Z.A.A, Joda F.A, and Naser M.A. (2023). A Survey on Detecting Abnormal Behavior Based on Gait Analysis. International Conference on Information Technology Applied Mathematics Statistics (2023), under submission in IEEE Xplore digital library.

2. Alhuseen, Z.A.A., Joda, F.A., Naser, M.A. (2023). Abnormal behavior detection in gait analysis using convolutional neural networks. Ingénierie des Systèmes d'Information, Vol. 28, No. 5, pp. 1151-1159. https://doi.org/10.18280/isi.280504

المستخلص

أبدى الباحثون مؤخراً اهتماماً واسع النطاق بتصنيف السلوكيات غير الطبيعية باستخدام الرؤية الحاسوبية، وذلك لتعزيز الأمن والسلامة والكفاءة في مختلف المجالات. ويواصل الباحثون استكشاف وتحسين التقنيات لجعل هذه الأنظمة أكثر فعالية وموثوقية.

ويأتي النظام المقترح كمحاولة لتصنيف أنماط المشية غير الطبيعية بناءً على تحليل المشية ومشاهد المشي، حيث يتم تحليل مشية الأشخاص في عدة مواقف. ولذلك فهو يعمل على تقديم نموذج للكشف عن الانحرافات في مشية الأفراد بناء على أنماط المشي.

يعمل هذا النظام على مجموعة من مقاطع الفيديو للبيانات التي تم جمعها للأشخاص الذين يعانون من أمراض معينة تؤثر على مشيتهم وغيرهم من الأشخاص الأصحاء، ويجري عليها عدد من العمليات المسبقة، ثم يطبق عليها عدد من تقنيات استخراج الميزات للوصول إلى تصنيف الصور، بهدف مقارنة وتقييم أداء هذه التقنيات، وهي: الرسم البياني للتدرجات الموجهة(HOG) ، والنمط الثنائي المحلي(LBP)، والنمط الثنائي المحلي المتماثل المركزي(CS-LBP) ، والنمط الثنائي المحلي المتماثل المركزي الممتد (XCS-LBP). وبعد الحصول على الميزات المستخرجة يتم الدخول إلى المرحلة الأخيرة والأهم في النظام وهي مرحلة التصنيف باستخدام إحدى خوارزميات التعلم العميق وهي(CNN+Vgg16) .

لتقييم كفاءة النموذج المقدم وإجراء التقييم على مجموعة متنوعة من مجموعات البيانات المجمعة بما في ذلك حالات السلوك الطبيعي وغير الطبيعي. حقق النظام أداء جيد وبدقة تصل إلى 99% باستخدام طريقتي CS-LBP و XCS-LBP .وتشير النتائج التجريبية إلى أن هاتين الطريقتين جيدتان في تسريع أوقات التدريب والاختبار، كما أظهرت التجارب أن طرق استخلاص ميزة المشية لعبت دوراً في نجاح النظام وفي تصنيف أنماط المشي غير الطبيعية.

وزارة التعليم العالي والبحث العلمي
جامعة بابل
كلية العلوم للبنات
قسم علوم الحاسوب

تصنيف أنماط المشي غير الطبيعية بالاعتماد على خصائص الصورة المكانية وتقنية التعلم العميق

رسالة

مقدمة الى مجلس كلية العلوم للبنات – جامعة بابل وهي جزء من متطلبات نيل درجة الماجستير في العلوم/علوم الحاسوب

من قبل

زينب علي عبد الحسين قمر

بإشراف

أ.د. محمد عبد الله ناصر

م.د. فنر علي جودة

2024

1445هـ