SCHOOL OF BUILT ENVIRONMENT, ENGINEERING AND COMPUTING

LEEDS BECKETT UNIVERSITY

# Advanced Face Detection Approach for Smart Surveillance Framework

By: Muhammad Noman

Supervisor Name: Dr Muhammad Shan-e-Khuda

Submitted to Leeds Beckett University in partial fulfilment of the requirements for the degree of MSc Data Science

**May 2024**

# Candidate's Declaration

I, Muhammad Noman, confirm that this dissertation and its work are my achievements.

Where I have consulted the published work of others, this is always clearly attributed;

Where I have quoted from the work of others, the source is always given. Except for such quotations, this dissertation is entirely my work;

I have acknowledged all primary sources of help;

I have read and understand the penalties associated with Academic Misconduct.

Signed:

Date: 08-05-2024

Student ID No: 77350149

# Acknowledgements

I want to express my sincere gratitude to my thesis supervisor, Muhammad Shan-e-Khuda, for their invaluable guidance, support, and mentorship throughout the research process. I am deeply thankful for their insightful feedback, encouragement, and dedication, which have been instrumental in shaping this thesis. I would also like to thank the faculty members of the LBU for their expertise and assistance. Their constructive criticism and suggestions have contributed significantly to the refinement of this work.

I am grateful to my family and friends for their unwavering encouragement, understanding, and patience during this journey. Their love and support have been a constant source of motivation and inspiration. I want to acknowledge the participants who generously contributed their time and insights to this study. Without their cooperation and involvement, this research would not have been possible.

This thesis is dedicated to all who have participated in its completion, no matter how small. Thank you for being a part of this journey.

# Abstract

This dissertation project proposes an advanced face detection system motivated by the collection of data science practices such as data analysis, data visualisation and prediction in surveillance and security. Our project initiated the thorough preprocessing of video-based datasets, an initial step to preparing the data for analysis. We ensured the data was optimized for accurate feature extraction techniques such as frame sequence conversion and noise reduction. This made it possible to maintain the framework for data science-oriented processes. Converting the data from preprocessing, our focus is extracting regions of interest (ROI) from images with the help of advanced data science techniques to extract facial features. This step improved the system's capability to concentrate on relevant features for further analysis. Feature extraction followed suit, retaining sophisticated procedures such as motion, change and distance features to capture facial appearances. These features were then set into feature vectors, allowing actual illustration and study of facial patterns for classification. Moving forward, the classification phase utilizes data science procedures, leveraging algorithms such as Multilayer Perceptron (MLP) to classify facial patterns. The MLP was trained to distinguish facial characteristics through supervised learning techniques and iterative parameter tuning, attaining high precision in classification tasks. Data optimization developed as an essential aspect of our project, where data science methods such as metaheuristic algorithms like Grey Wolf Optimization (GWO) were utilized to fine-tune feature weights and improve system efficiency. Through iterative modification, the system maintained the accuracy and robustness of video environments, showcasing the power of data-driven optimization in real-world tenders. Throughout our project, we face various challenges integral to data science activities, including environmental inconsistency, resource limitations, ethical considerations, and limitations in generalization. Despite these challenges, the developed face detection system grasps huge potential in various data science applications. From security and surveillance systems to automated attendance tracking, customer analytics, and healthcare, the system offers criminal insights and decision-making abilities through data science.

# Contents

# List of Figures

# List of Tables

# Chapter 1: Introduction

## 1.1 Overview

In today's swiftly changing ecosphere, the importance of improving surveillance systems has always been more apparent, specifically with the development of Advanced Face Recognition technology. This motivation highlights the requirement for a paradigm-shifting combination of state-of-the-art facial recognition procedures to allow a more complex and efficient method in intelligent surveillance applications (De-La-Torre et al., 2015). This research is motivated by the recognition that traditional methods are utilized to maintain the demands of modern surveillance. To meet our increasing requirements for precision and flexibility, we have reached an interval wherein more intelligent and effective systems are required (Sabir et al., 2019). We aim to renovate how we monitor and protect our environments by integrating advanced facial recognition techniques into surveillance applications. The research was motivated by the complexities of contemporary surveillance, which made it challenging for conventional approaches to fulfil the changing needs for accuracy, effectiveness, and flexibility for modern systems(Chua et al., 2002).

In modern biometric systems, face detection is a critical module that supports security and surveillance applications primarily declaring on video-based datasets. This research methodically investigates using data science methodologies to develop a robust face detection structure designed clearly for video-centric datasets. The process starts with systematically creating data from video sources, essential in providing the necessary information for the following analysis. Videos are divided into separate frames with accurate frame conversion, preparing them for an in-depth analysis(Li et al., 2020). The transformation of these frames results in the extraction of Regions of Interest (ROI), which are facial areas of significance. Through employing advanced algorithms that include skin, edge, and change detection, these regions of interest (ROIs) are reasonably defined, which helps as a critical foundation for subsequent analysis.

After the process of ROI extraction is complete, the next step is feature extraction, which includes reducing the core of facial features to different attributes(Chen and McGurr, 2016). These characteristics, which contain texture, form, and type, form the substance for differentiating unique facial patterns. Feasible structures' complex details are captured and restrained via engaging feature extraction methods with the decision, including Histogram of Oriented Gradients (HOG) and Local Binary Patterns

(LBP) and other features such as motion, change and distance. The feature vectors obtained are systematically arranged and transferred into a structured format, commonly a CSV file. This procedure ensures a smooth combination of the feature vectors into subsequent processing stages with valuable information(Chen et al., 2014).

After the datasets and feature vectors have been carefully chosen and constructed, the attention moves to the optimization phase. This stage is crucial in enhancing the effectiveness of the face detection structure. Grey Wolf Optimisation (GWO) creates itself as a challenging adversary by implementing optimization methodologies that do not incorporate artificial intelligence(Montserrat et al., 2020). Motivated by the challenging hunting capabilities and composite social structure of grey wolves, GWO skillfully crosses the complex territory of parameter optimization to expose the algorithm's dormant possibility in the face detection domain. In engaging an iterative modification procedure, GWO improves the system's adaptability to various video environments, optimizes performance metrics, and fine-tunes algorithmic constraints(Viola and Jones, 2004). As the optimized algorithm for face detection approaches implementation, classification appears to be the ultimate domain in vision. Prepared with rich feature vectors and boosting the robustness of machine learning techniques, the Multilayer Perceptron (MLP) is used to indicate its potential for adequate classification. In a divergence from predictable artificial neural networks, MLP services an incrusted architecture that allows the achievement of knowledge regarding complex forms in the feature space. Following a problematic training process incorporating backpropagation and gradient descent, the MLP capably identifies the gentle distinctions of facial features, eventually producing a structure for categorical classification.

This research responsibility demonstrates a comprehensive face detection methodology founded upon smoothly incorporating data science views and technical expertise(Dong et al., 2020). A robust face detection pipeline is formed by coordinating the processes of data generation, feature extraction, optimization, and classification, which are capable of effortlessly navigating the intricacies inherent in video-based datasets(Korshunov and Marcel, 2018). As the trajectory progresses, the detection of improved security, surveillance, and biometric systems is the guiding principle, driving the development of face detection technologies into unknown domains of effectiveness and reliability(Gupta and Tiwari, 2014). Despite their initial innovative nature, predictable facial recognition methods have encountered challenges that have needed a paradigm shift(Zhou et al., 2017). The limitations incorporate a range of difficulties associated with being specific, enhancing efficacy, and adapting to diverse environmental conditions. (Kumar et al., 2018) state the following. Swift facial recognition is difficult to achieve in intelligent surveillance systems. A current movement has been towards incorporating machine learning techniques to address these constraints(Jiang and Learned-Miller, 2017). The implementation of machine learning algorithms, particularly employing feature engineering and data optimization, leads to the improvement of conventional limitations. (Wang, 2022)Subsequently, this study aims to advance face recognition in intelligent surveillance systems by creating a novel and reliable method(Bonettini et al., 2021). The proposed face identification system aims to decrease the computational expenses of real-time recognition and enhance precision and efficiency. Integrating machine learning techniques aims to improve recognition speed, accuracy, and computational efficiency(Jiang and Learned-Miller, 2017). The focus on reducing computational costs is consistent with optimizing resources (Mallick et al., 2020). The increasing frequency of intelligent surveillance systems requires the development of resource-efficient, high-performance solutions (Beyan et al., 2022).

The primary objective of this study is to address the growing need for efficient, intelligent monitoring methods. The aim is to enhance face detection by directly integrating machine learning methodologies. It has enhanced efficiency results from this integration, which offers a sustainable, adaptable solution to the ever-changing demands of intelligent surveillance.

## 1.2 Rationale

Surveillance systems are significant in ensuring safety and security in the modern world. Current face recognition methods have various constraints when precisely discerning objects, mostly in complex video datasets. Traditional methods face some challenges, such as shape and motion, light conditions and other issues in various videos, which decrease the efficacy and performance. To reduce face detection problems, there is a collective need to use contemporary face recognition techniques, particularly ones that use machine learning. Researchers aim to enhance the reliability of facial recognition systems throughout distinct sceneries by advanced methods and algorithms. The motivation for leading this research is derived from a critical need for an enhanced method to overcome the challenges. Traditional approaches have issues in providing accuracy; in this regard, there is a need to develop a robust system that improves the systems' accuracy, reduces computational costs, and provides reliability.

By enhancing the capabilities of surveillance systems, attaining this objective would enable more effective monitoring and detection of security concerns. Incapacitating these challenges and improving the characteristics of face recognition technology, we will establish the basis for more secure and preserved individuals. The primary aim of this research activity is to provide a scholarly contribution to the structure via novel methodologies that could update current surveillance techniques.

**Impact:** The research seeks to propose a robust, optimized, and high-performance solution.

## 1.3 Aim and Objectives

The proposed solution aims to achieve a robust and effective method to detect faces and objects, extract the hand-crafting features, reduce the dimension with optimization and finally classify the data. This system can be applied to various domains, such as surveillance, sports, and other intelligent systems.

## 1.4 Outline

This dissertation is organized as follows:

Chapter 2 explores the literature review of the project, including the main related work and the limitations and challenges of the existing and updated literature. In literature, we use a thematic approach for every research study and highlight the contributions, scheme of research and limitations of the papers.

Chapter 3 defines the detailed methodology and proposed solution; in this, we describe the details of the project, how we collect the dataset, its link and definition, preprocessing method, features engineering, dataset comma separated file (CSV) making, discussion of the results and algorithms are also mentioned.

Chapter 4 delivers the product/research design and implementation, having the idea of how we consider or apply our project as a product for society and discussing the research design and implementation of the project.

Chapter 5 describes the research outcomes/results/discussion and evaluation; in the outcomes section, we discuss the outcome of the project, in the discussion, we provide the informal details of the project,

Chapter 6 discusses the project management method and procedures used in this research project, which method we have used and what will be the impact of this project; every technique has its properties; we applied the agile method to utilize units and complete every unit in a given time.

Finally, Chapter 7 describes the conclusion and future work.

## Summary of the Chapter:

This chapter discusses the main idea and motivation of the project, the research domain, how to implement this project in other domains, the project's impact, related examples, and the organisation of the report.

# Chapter 2: Literature Review

In arranging the basis for our research into advanced face detection methods, we initiate a thematic literature review, researching the multilayered landscape of modern intelligent reflection. Our expedition was directed through critical themes, including facial recognition examples, machine learning applications, and the addition of advanced technologies within the modern technologies of surveillance systems.

In creating the framework for exploring advanced face detection methodologies, we get a comprehensive thematic literature review, researching the complex domain of modern intelligent surveillance. Fundamental themes guided our journey about practical applications of facial recognition, the utilization of machine learning algorithms, and the continuous integration of pioneering technologies within current surveillance systems. Navigating these fundamental areas provided invaluable insights into the complicated landscape of facial detection and its symbiotic relationship with state-of-the-art surveillance technologies. By submerging ourselves in the depth of these topics, we are required to gain a basic understanding of the field's current state and identify emerging trends and challenges. Through this review process, we can establish a robust foundation of knowledge that would serve as a managerial compass for our research product. Through various industrial perspectives and drawing upon insights from interdisciplinary foundations, we aimed to chart a course toward developing innovative solutions that address the evolving demands of intelligent observation in the digital age. Expanding upon this thematic exploration improves our understanding of the complex relationship between facial detection and surveillance technologies and provides a roadmap for future research directions. By researching more deeply into the distinctions of each theme and clarifying their connection, we try to expose novel insights that boost the field forward and cover the way for transformative advancements in intelligent surveillance.

## Paper 01: A Real-Time Framework for Human Face Detection and Recognition in CCTV Images Authors: Rehmat Ulla et al.

***Thematic Overview:*** Advancements in Real-time Face Recognition: This theme explores the progress of real-time face recognition methods, researching the integration of machine learning and deep learning techniques. The aim is to overcome challenges modelled by differences in lighting, rotation, scaling, and cluttered backgrounds within the CCTV images(Ullah et al., 2022). Feature Extraction Techniques: This theme focuses on the essential role of the feature extraction approach, specifically Principal Component Analysis (PCA) and Convolutional Neural Networks (CNN). The argument navigates over their applications in improving recognition accuracy by capturing essential facial features in complex conditions.

***Algorithmic Evaluation:*** This theme critically measures key algorithms such as K-Nearest Neighbors (KNN), decision trees, random forests, and CNNs. The study contains their fitness for real-time face recognition in CCTV images, seeing factors like efficiency, accuracy, and adaptability to challenging scenarios.

***Technologies Involved and Limitations:*** Advanced technologies are utilized in the search's effort to develop an environment for real-time face recognition on CCTV images. Integrating machine learning and deep learning methods with methods for extraction of features such as PCA and CNNs, these techniques form the basis. The framework comprises image acquisition, preprocessing, face detection, localization, feature extraction, and recognition procedures. Nevertheless, inherent constraints are recognized. Limits are facing the field of image-based recognition, such as challenges such as light variations, scaling, rotation, and congested backgrounds. In managing complex scenarios, the search for reduced human involvement and cost-effectiveness will be vulnerable, which could compromise optimal precision under certain conditions. With these challenges, the study attempts to achieve a harmonious symmetry between technological progress and the pragmatic limitations of CCTV images from the real world.

## Paper 02: Live Detection of Face Using Machine Learning with Multifeature Method

***Authors: Sandeep Kumar et al.***

***Thematic Overview:*** Advancements in Facial Expression Detection Algorithms: This theme highlights the development and improvement of an algorithm for automatic live Facial Expression Detection (FED) using advanced methods such as radial basis function, Haar discrete wavelet transform, and Gray-level difference method(Kumar et al., 2018). Innovative Methodologies for Feature Extraction and Classification: The research explores using the Otsu algorithm for detecting edges in facial images, with an innovative method to improve the accuracy and efficiency of feature extraction.

Applying feature extraction and classification framework on diverse databases such as Japanese Female Facial Expressions, Cohn–Kanade Extended (CK+), CMU, BioID, Long Distance, and FEI contributes to an inclusive consideration of facial expression detection.

Dataset Diversity and Evaluation Strategies: This theme highlights the implication of deploying a variety of datasets, including FEI, Japanese Female Facial Expressions, Cohn-Kanade Extended (CK+), CMU, BioID, and Long Distance. This indicates the flexibility and adaptability of the method in a wide variety of conditions. The study uses strategies such as contrast improvement, median filtering, and learning-to-rank-based methods to solve constraints and enhance the algorithm's efficacy.

***Technological Insights and Limitations:***

Integrating the grey-level variance method, radial basis operation, and Haar discrete wavelet transformation into its scientific framework, the real-time facial expression recognition algorithm deploys an integrated approach. The integration provides inclusive feature extraction and classification by utilizing multiple databases and the Otsu algorithm, which is used for edge detection. The study recognizes particular constraints and moderates them by implementing long-distance and low-contrast images instead of the initial algorithmic restrictions and resolution improvements. Feature extraction is refined via median filtering during the preprocessing phase. The analysis methodology joins in learning-to-rank-based methods and attains the highest expression recognition rate of 100%. This achievement showcases the model's exceptional precision and recall.

## Paper 03: Face Detection in Security Monitoring Based on Artificial Intelligence Video Retrieval Technology

***Authors: ZUOLIN DONG et al.***

***Thematic Overview:*** Developing Face Detection for Videos: This theme contains the overview of an emerging robust video-based face detection approach with deep learning features. The main objective is to improve face detection abilities in a dynamic video situation (Dong et al., 2020). Integration of Video Processing and Deep Learning: This research combines video processing, computer vision, and artificial intelligence to produce an effective monitoring system. A deep learning network includes edge, contour, local, and semantic features.

**Validation in Real-world Scenarios:** This theme highlights the significance of assessing the algorithm's applicability in real-world scenarios. Despite the algorithm's impressive accuracy, exploring diverse situations to pinpoint potential challenges is crucial. The study weighs the need to meet the demands of real-time face detection and suggests establishing comprehensive criteria to enhance the algorithm's scalability. By researching various scenarios, researchers can uncover limitations and refine the algorithm to ensure optimal performance across different environments and conditions. This proactive approach promotes the development of robust and versatile algorithms capable of meeting the diverse needs of users in practical settings.

With the validation of the algorithm with data from the Face in Action dataset, researchers can illustrate its practical implementation and effectiveness in real-world applications. This validation process is crucial in demonstrating the algorithm's reliability and suitability for deployment in various scenarios, ranging from surveillance to intelligent systems. Through meticulous validation and testing, researchers can ensure that the algorithm meets the rigorous requirements of real-world applications, including accuracy, efficiency, and adaptability. The proposed approach to memory management, utilizing Kullback-Leibler divergence for sample selection, represents a significant advancement in optimizing the adaptability of the entire system. Researchers can improve the system's ability to select relevant samples via innovative techniques, enhancing its performance in dynamic and evolving environments. This approach enhances the algorithm's adaptability, efficiency, and effectiveness in real-world scenarios. The study emphasizes the importance of testing algorithms in practical settings to identify potential challenges and refine their performance. By validating the algorithm with real-world data and proposing innovative approaches to memory management, researchers can enhance the algorithm's scalability, adaptability, and effectiveness in addressing the diverse needs of users. Through continuous validation, testing, and refinement, researchers ensure that algorithms meet the demands of real-world applications, paving the way for their successful deployment in various domains.

***Technological Limitations:*** The scholarly framework of the technique is outstanding for its revolutionary combination of computational intelligence, video processing, and computer vision. Using edge, shape, local, and semantic features, an advanced deep learning network is constructed to assure consistent face detection, irrespective of any challenges. While the study recognizes particular limitations and highlights the importance of further validation elsewhere modeling to verify execution in real life, it also addresses feasible challenges that could develop in various scenarios, regardless of the algorithm's acceptable result. Establishing specific requirements to measure the algorithm's ability to identify faces in real time is critical. To perform this, a comprehensive framework should be created for enabling technological advancement.

## Paper 04: Paper Name: Partially-supervised learning from facial trajectories for face recognition in video surveillance.

***Authors: Miguel De-la-Torre et al.***

**Thematic Overview:** Face Recognition Systems: This theme covers the design of an adaptive Multiple Classifier System (MCS) for video surveillance regarding face identification. The main objective is to develop and implement an architecture that can respond flexibly to developing facial representations and paths over time, with the final aim of enhancing spatiotemporal recognition in video data. (De-La-Torre et al., 2015). Facial Trajectories and Spatio-Temporal Recognition: The development of deploying facial paths inside the structure of partially supervised learning of facial methods is the principal objective of the research approach. A robust spatio-temporal identity is the aim of the apparatus, which uses person-specific groups and data from a face detector. A learn-and-combine method minimizes the probability of data bias during the ensemble's self-updating method.

***Memory Management and Sample Selection:*** An approach to memory management that uses Kullback-Leibler separation to optimize sample selection, consequently improving the adaptability of the entire system, is proposed. The dataset is validated by utilizing the Face in Action dataset records that demonstrate the practical implementation of the developed adaptive MCS. They proposed an approach to memory management that utilizes Kullback-Leibler separation to optimize sample selection, thereby enhancing the adaptability of the entire system. This method suggests retaining only the most relevant samples in memory, thereby reducing computational overhead and improving system efficiency. The system can dynamically adjust its memory usage to accommodate changing conditions and requirements by selecting samples based on their divergence from the current model.

They used records from the Face in Action dataset to validate their approach, which provided real-world examples of facial terminologies and actions. By applying their adaptive memory management technique to this dataset, they demonstrated its practical implementation and effectiveness in improving system performance. The dataset was valuable for testing and validating their approach under diverse conditions and scenarios. Their proposed method offers a flexible and scalable solution for memory management in machine learning systems, particularly those dealing with dynamic environments and evolving datasets. Kullback-Leibler separation could effectively balance the trade-off between memory usage and model accuracy, ensuring optimal application performance. Their approach contributes to the advancement of memory-efficient algorithms and reinforces the importance of adaptability in modern computing systems.

***Technological Insights and Limitations:*** The technological structure of interactive spatiotemporal identification uses facial trajectories, face tracking data, and ensembles specific to every subject. While self-update processes arise, the learn-and-combine strategy is key for avoiding information bias. An additional factor enhancing the system's adaptability is a memory management approach predicated on theKullback–Leibler deviation, which facilitates the selection of related samples. The study recognizes constraints and supports additional experimentation in various practical situations to authenticate the proof of concept. In certain circumstances, the optimal adaptation to all pertinent target trajectories might exhibit superior performance to that of the proposed system. Finally, applying self-updating ensembles to individuals is contingent upon particular scenarios.

## Summary of the Chapter:

This chapter discusses the related work of the project; we consider the primary four papers as related work, a discussion is started and a thematic summary of the project, main contribution of the paper and challenges which researchers have faced during the research; finally we discuss the citation of the project and explore the research gap.

# Chapter 3: Methodology

In this chapter, we discussed our proposed method. Initially, we considered video-based data to be over-input. The next step is data preprocessing, which is essential for image-based data to improve the performance of machine learning models. Initially, we convert video data into frame sequences and apply a resizing algorithm to resize the image. After this, the next step was noise reduction in a given image set; video-based data contain motion blur noise; we apply a median filter to reduce this noise. Images mostly contain unwanted artifacts; after noise reduction, the next step was to extract the region of interest (ROI); ROI is the key region, to achieve roi we applied various methods to reach the goal, including multiple algorithms such as Skin tone detection, Edge detection and motion detection. Finally, we optimized all these results to get the final results. After this, we have the features extraction phase, where we extract various features to get information about hidden patterns. The next step is data optimization and applying the classification method to get information about predictive results. Figure 3-1 shows the procedure of the proposed method.

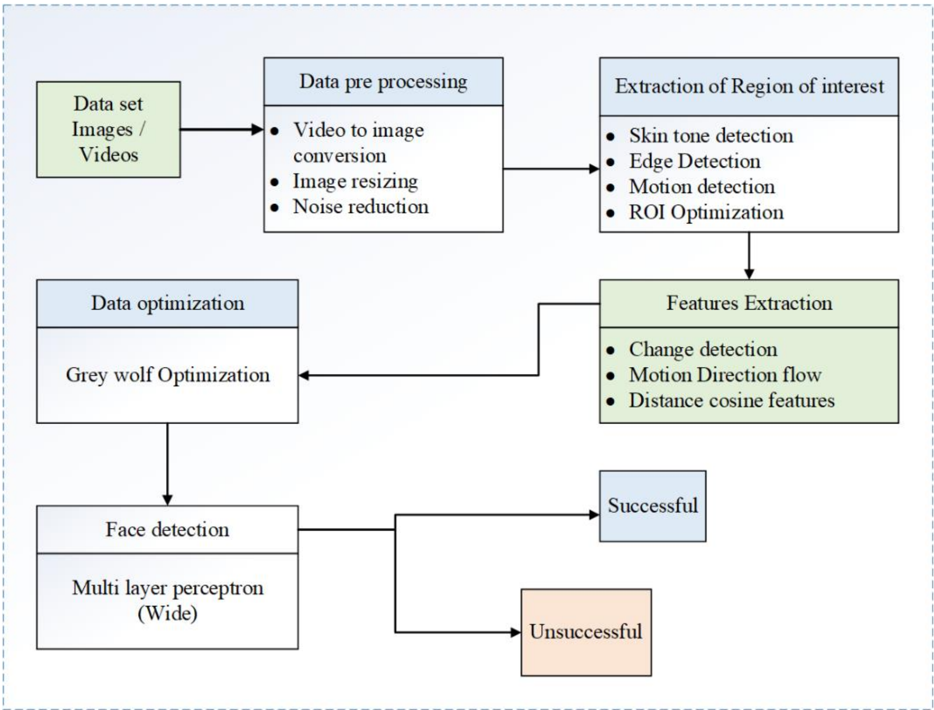


*Figure 3-1: The model diagram of the proposed framework.*

## 3.1 Ethical Considerations

The proposed research aims to enhance face detection techniques for intelligent surveillance systems in complete compliance with ethical principles highlighting privacy, consent, and responsible technology procedures. The primary goal is to improve the precision and effectiveness of facial recognition technologies within surveillance situations. The methodology of the project integrates ethical considerations systematically, confirming the maintaining of privacy during every aspect of data processing, including noise reduction and region of interest extraction.

Feature extraction methods, such as distance cosines, motion detection, and motion direction flow, have been designed to extract appropriate data for facial recognition while providing a sense of privacy. Grey Wolf Optimisation (GWO) and Multilayer Perceptron (MLP) are illustrations of machine learning algorithms applied to improve system performance while maintaining ethical procedures. Clear and unbiased data preprocessing methods, such as missing and replicated values, are employed to sustain principles of fairness of data.

Integrating GWO as an optimization algorithm validates a dedication to ethical AI methods, which helps transparency and integrity. This research maintains every aspect to uphold ethical principles, employing responsible skills in data science and surveillance. Ensuring ethical standards, the methodology and algorithms protect the privacy and individual rights throughout the research process. The project's assurance to enhance surveillance technology is compatible with its ethical considerations and follows the principles of responsible AI expansion.

## 3.2 Data set:

An avenue dataset contains a total of 15 sequences. Each sequence takes approximately two minutes(Lu et al., 2013). There are 35,240 frames in total. Four videos are used as training data for 8,478 frames among the fourteen unusual rates. Figure 3-2 illustrates the example frame of the avenue dataset.

*Figure 3-2: The example images of the dataset*

## 3.3 Data Preprocessing:

Data preprocessing is the primary step to avoid extra computational cost, memory use, and time optimization. In this research, we apply three steps for data preprocessing, such as video-to-frame conversion. After that, image resizing is very important for performing uniform processing(Ghadi et al., 2022a). The next and final step is noise reduction; due to motion and other parameters, every video has some extra information we deal with as noise. We apply the noise reduction filtration process to get desirable results. Algorithm 1 shows the complete details of preprocessing farmwork.

*Table 1: Preprocessing of the input data*

| **Algorithm 01:** Preprocessing of the input data |
|---|
| *Input:* |
| Video-based data |
| *Video to frame conversion:* |
| Set a rate and convert all video into frames. |
| *Frame resizing:* |
| Adjust a uniform size for all frames. |
| *Noise reduction:* |
| Apply motion blur noise to reduce the noise. |
| *Output:* |
| Return the solution provided in the next frame. |

### 3.3.1 Video to Frame conversion:

This section discusses video-to-frame conversion as a fundamental preprocessing step to simplify the complete study and understanding of video content(Yao et al., 2016). In performing frame-by-frame conversion of video data, an improved evaluation of visual information is attained, which is helpful in various tasks, including face detection, motion analysis, and prediction(Alam et al., 2022). The procedure entails sequentially extracting frames from the video source or resizing frames to improve their quality. The primary information for working models and techniques used in the study consists of these extracted frames. The mathematical description of the conversion from video to frames is as follows:

$$Frames() = Vid(\times f\_r),\ Frames(t) = Vid(t \times f\_r) \tag{1}$$

where $Frames()$, $Frames(t)$ denotes the frames extracted from the video at time $t$, and Vid $(\times\ f\_r), Vid(t \times f\_r)$ represents the video sampled at time $t$ reproduced by the frame rate to determine the equivalent frame. The goal is to utilize the considerable volume of content that exists in video data to generate insightful analyses and significant outcomes.

### 3.3.2 Image Resizing:

In the section where we discuss image resizing, we implement a standardized approach to ensure consistency across all extracted frames. By selecting a consistent height and Width for resizing, which is $Height = 240$ and $Width = 320$ pixels (Akhter and Jalal, 2023). The preprocessing procedure is optimized for successive evaluation by implementing uniform resizing(Aljuaid et al., 2023). Which allows for smooth integration into the following segments through allowing comparison and computation. The resizing method is mathematically illustrated as

$$Re_F = imre(Or_F, [240{,}320])Re\_F = imR(Or_F, [240{,}320]), \tag{2}$$

where $Res\_F$ denotes the resulting resized frame, and $Or\_F$ represents the frame extracted from the video source. By implementing this uniform resizing method, later algorithms and statistical methods can function more efficiently and frequently, improving the research findings and increasing reliability. Figure 3-3 illustrates the results of resizing the frame.


*Figure 3-3: The example resize images*

### 3.3.3 Noise Reduction:

In our video processing pipeline, noise reduction enhances the quality and accuracy of succeeding analyses (Ghadi et al., 2022b). To achieve this, we employ a median filter, a common technique for effectively reducing various types of image noise. By applying the median filter to each video stream frame, we aim to suppress noise while preserving essential image features (Akhter and Javeed, 2022). We employ the median filter with an appropriate kernel size to reduce noise-induced anomalies and inconsistencies. This process generates better frames. The equation for the noise reduction procedure is

$$Fil_F = medfilt2(OriF, [,])Fil_F = medfilt2(OriF, [m, n]), \tag{3}$$

where $Fil\_F$ represents the frame after noise reduction, $OriF$ denotes the frame extracted from the video, and $[m, n]$ specifies the size of the median filter kernel. This method enhances the quality of the video data, which is facilitated with greater precision, as noise is significantly reduced. Figure 3-4 presents the comprehensive outcomes of the image filtering procedure.


*Figure 3-4: The results of filtered images*

## 3.4 Extraction of Region of Interest:

After the initial phase, the region of interest (ROI) extraction is required. Our target area of focus is the (ROI), and we utilize additional techniques to extract the hidden data pattern. This procedure consists of four stages: first, skin tone detection is used to extract information about the skin in nearby areas of the human face; second, edge detection is implemented to identify the boundaries of the frames and relevant data; third, motion detection is performed; and finally, the optimization process is employed(Akhtar et al., 2023; Ghadi et al., 2021; Mehmood et al., 2019). The second algorithm demonstrates the serial procedures of extracting the region of interest.

*Table 2 Extraction of the region of interest.*

**Algorithm 02:** Extraction of the region of interest.

*Input:*

Frame data

- *Skin Tone detection:*
  Apply the skin tone detection method and map all frames in the separate data frame
- *Edge Detection:*
  Apply edge detection procedure in the denoise frame and map app of them in the different data frame
- *Motion Detection:*
  Extract the motion from denoise frames
- *ROI Optimization:*
  Add and optimize all the above steps to get better results.

*Output:*

Return the solution provided in the following data frame.

### 3.4.1 Skin Tone Detection:

Skin tone detection, a key part of several applications, including face recognition, gesture assistance, and emotion evaluation, is executed within this segment. To achieve this, skin tone detection algorithms are applied to the filtered frames to identify areas similar to human skin. By segmenting the skin tones from the background and other elements visible in the frame, it is possible to significantly improve the precision and dependability of subsequent algorithms(Cheddad et al., 2009). To differentiate skin pixels from non-skin pixels, the skin tone detection procedure involves applying color space changes, machine learning and statistical modelling. The mathematical representation of the skin tone detection operation is as follows:

$$SkM = SkTD(FiF) \tag{4}$$

The binary mask denoted as $SKM$ indicates the existence of skin pixels within the filtered frame and $SkTD(FiF)$, shows the duncation on the given frame. This methodology allows for the separation and extraction of relevant information regarding human subjects, thereby promoting analysis that is more precise and context-active for various areas of research. The results are illustrated in Figure 3- 5.


*Figure 3-5: The results of filtered images*

### 3.4.2 Edge Detection:

Edge detection performs a critical role in this procedure as it determines and indicates the boundaries of objects within frames. To accomplish this, the Canny edge detection algorithm is widely recognized for its resilience and efficacy in the edge detection of images(Gochoo et al., 2021). Each frame is initially converted to grayscale to streamline edge detection operations. Following this, the grayscale frame is subjected to the Canny edge detection algorithm, which emphasizes and identifies regions of substantial intensity variation that commonly correspond to the edges of objects in the scene(Canny, 1986). The mathematical expression for the edge detection operation is

$$Edges = Canny(GrF, th1, th2)Ed \tag{5}$$

where $Edges$ represents the edge-detected frame, $GrF$ means the grayscale form of the original frame, and $th1$ and $th2$ are thresholds controlling the sensitivity of the edge detection process(Pervaiz et al., 2022). This methodology allows for the visualization and study of the boundaries of objects, which improves a wide range of research tasks, including object recognition, tracking, and motion analysis. Figure 3-6 illustrates the results of the edge detection process.

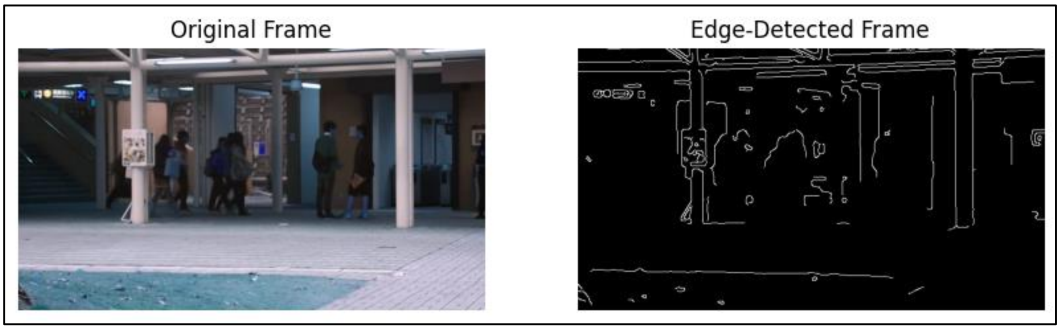


*Figure 3-6: The results of filtered images*

### 3.4.3 Motion Detection:

Movement detection within subregions is an essential step of our study as it enables the identification of regions of movement. To accomplish this, we utilize a motion detection algorithm that analyzes consecutive combinations of frames to identify pixel intensity variations of motion(Jalal et al., 2020). After converting each frame to grayscale, significant shifts are determined in regions by computing the total difference among consecutive frames (Ojala et al., 2022). A thresholding procedure next gives the variation in the image to categorize pixels as moving regions. The motion detection operation is  mathematically depicted as

$$MoD = Th(abs(CF - PF), th), MoDi = Th(abs(CF - PF), th) \quad (6)$$

where$MoD$ represents the resulting motion-detected frame $MoDi$, represents the resulting motion detection in the next frame $CF$ and $PF$ denote the current and previous grayscale frames, correy, and $th$ is the user-defined threshold controlling the sensitivity of the motion detection process. Figure 3-7 shows the results.


*Figure 3-7: The results of filtered images*

### 3.4.4 ROI Optimization:

For complete evaluation and understanding of the content of each frame, multimedia processing systems combine various detection techniques. Motion, edge, and skin tone detection algorithms are initially applied to the filtered frames acquired from the median filtering phase (Wibowo et al., 2021). In motion detection, regions of essential pixel intensity variations are identified by comparing consecutive frames; these differences serve as indicators of possible motion. The operation employing motion detection is expressed as

$$MoD = AmD(Fr, Fr + 1), MoDi = AmD(Frm, Frm + 1) \quad (7)$$

The Canny edge detection algorithm is employed to conduct the edge detection, consequently illustrating the existence of edges and boundaries within the frames. The operation for edge detection is described as

$$EdD = Canny(GrF, thr1, thr2), EdDi = Canny(GrF, thr1, thr2) \quad (8)$$

Skin tone detection aims to identify specific areas of human skin within the frames. This method separates skin pixels from the background via various statistical modelling techniques and color space adjustments. The process of detecting skin tone can be mathematically illustrated as

$$SkD = AsD(Frame), SkDi = AsD(Frame) \quad (9)$$

After gathering the results of the human analysis, a full depiction of the frame is achieved. The combined frame offers an entire scene view by integrating motion, edge, and skin tone detection information. This combined result frame simplifies subsequent things of analysis and interpretation. For variable check, reference equations are 4-6.

## 3.5 Features Extraction:

After the ROI extract, we apply the features extraction method to change detection, motion direction flow, and distance cosine features in the given data, and we prepare a dataset file for data optimization and classification. Algorithm 3 shows the map and guidelines for feature extraction.

*Table 3:* ***E****xtraction of Features.*

| **Algorithm 03:** Extraction of Features. |
| --- |
| *Input:* |
| ROI based frame |
| • *Change detection:* Apply the Change detection method and map all frames in the separate data frame |
| • *Motion Direction flow:* Extract the motion direction ROI frames |
| • *Distance cosine features:* Find the hidden patterns such as distance cosine. |
| *Output:* |
| Return the solution provided in the following data frame. |

### 3.5.1 Change detection:

Change detection is an essential method for extracting features as it allows for recognizing dynamic sections that occur among consecutive frames. The technique requires comparing two frames to identify regions with major variations in pixel intensity, which suggest adjustments to the scene or the motion of objects. To achieve this, the change detection algorithm is employed; it calculates the absolute difference between the pixel values of two grayscale frames. The change detection operation is mathematically depicted as

$$Change = abs(Fr1 - Fr2)Ch = abs(Fr1 - Fr2) \tag{10}$$

where $Fr1$ and $Fr2$ denote the grayscale illustrations of the original frame and the selected frame, $abs$ denotes the absolute values. A thresholding procedure is implemented to distinguish between unaltered and altered pixels in the total variation image. The change-detected frame indicates the areas of considerable variation, thereby offering significant insights into the dynamics of the scene and assisting in subsequent analysis methods, including object tracking, event detection, and anomaly recognition. Figure 3-8 illustrates the outcomes of the change detection process.


*Figure 3-8: The results of filtered images*

After getting the change detection features, we map them into the vector and Figure 9 shows the results of the mapped vector. Furthermore, we have used this feature vector and the rest of the two data optimization features to get better results.

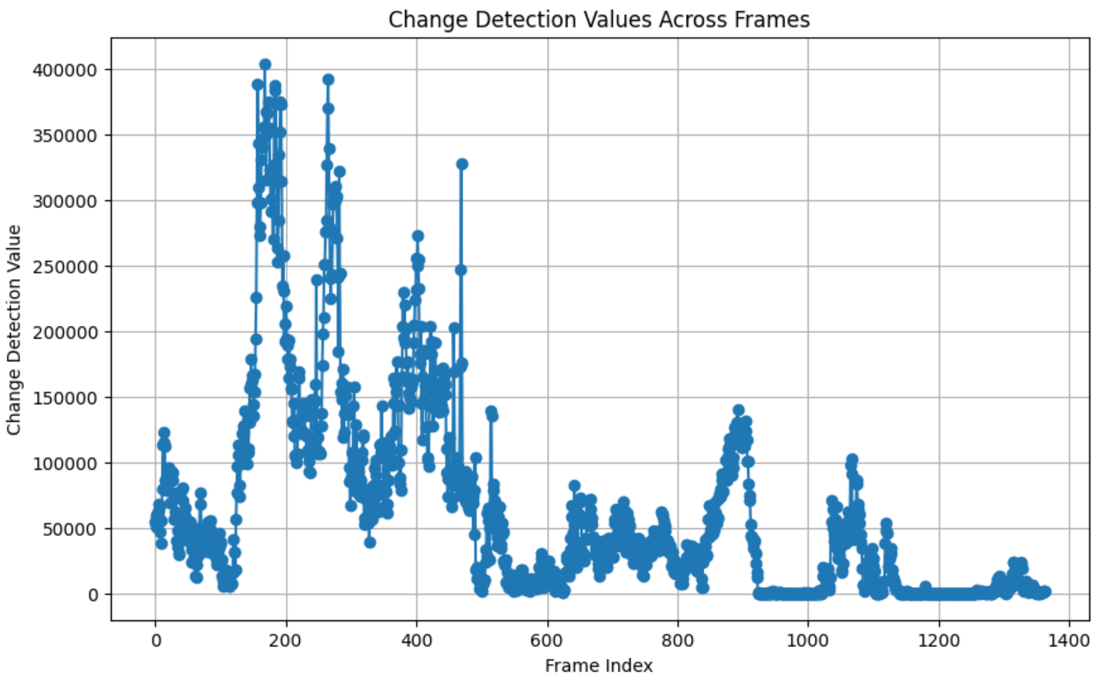


*Figure 3-9: The Graphical representation of change detection features*

### 3.5.2 Motion Direction Flow:

Motion direction flow is one of the main features of this research project. Using the direction of motion, we can find the values and direction points of the motion. We apply the mathematical formula i.e

$$mot \rightarrow (xi, yi) \tag{11}$$

Where $mot$ is the motion variable and $xi$ and $yi$ are the direction points. the $i$ and $j$ are the number of frames in a given data array; we extract all the values and map them into a vector. Figure 3-10 shows the graphical representation of the motion direction flow vector.

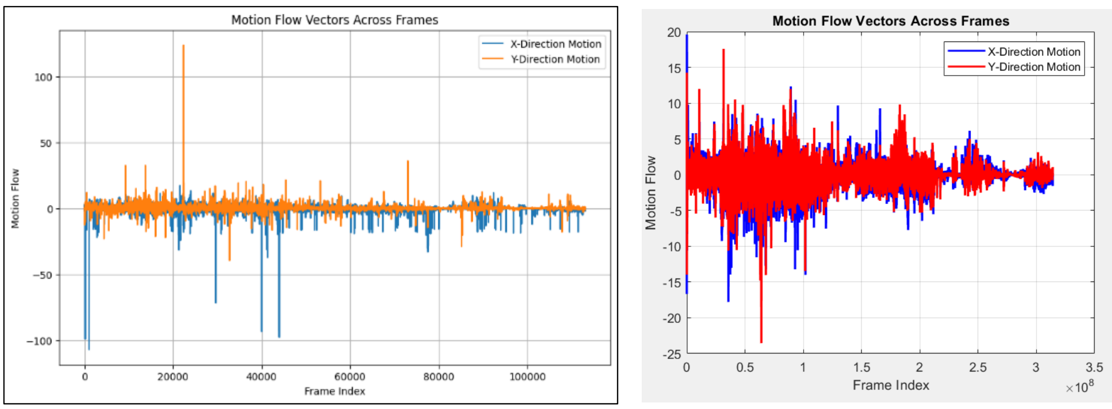


*Figure 3-10: The results of the Motion flow vector*

Figure 3-11 shows the average motion flow details.

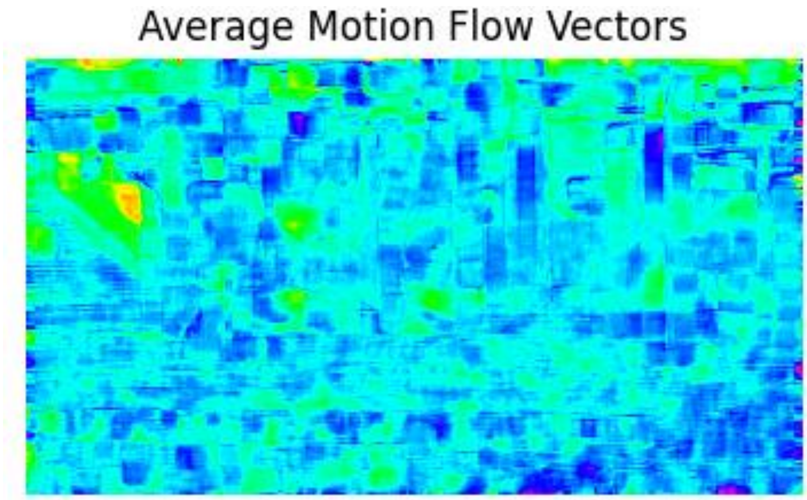


*Figure 3-11: The results of the Motion flow vector*

### 3.5.3 Distance cosine features:

The following features vector is distance cosine features, in which we target only the distance between existing objects in given frames and calculate the distance between them. Equations 12 show the mathematical representation of distance cosine features.

$$Dis = (obj_I \rightarrow A, Obj_J \rightarrow A) \quad (12)$$

Where $dis$ denotes the distance values, $obj$ is the object and $A$ denotes the object's angle, the $i$ and $j$ is the number of objects in the given data array. Figure 12 shows the mean distance features results.

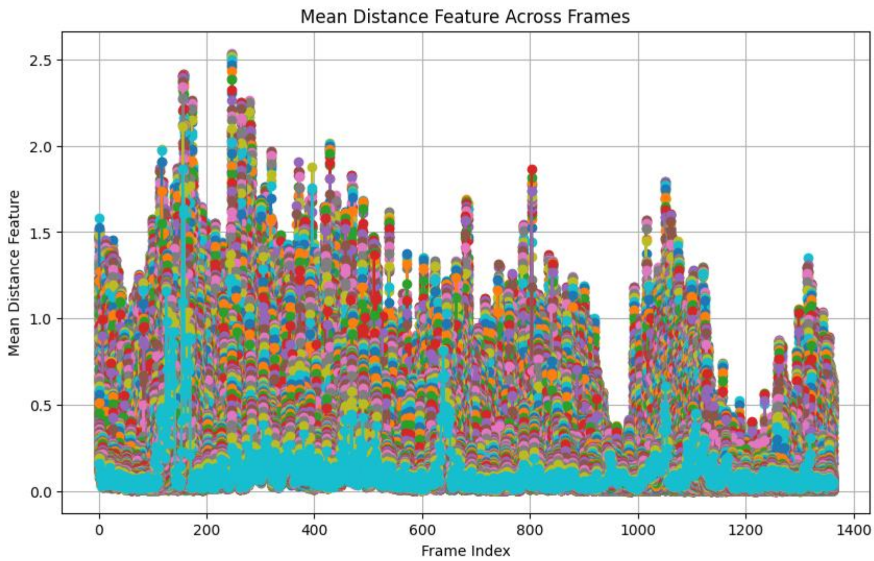

*Figure 3-12: The results of Distance features*

Figure 3-13 shows the shape of all feature vectors; using the normalization method, we applied the same shape mechanism for making the same shape. It will help us with data optimization and classification.

```
Shape of Change Detection Vector: (314035200,)
Shape of Motion Flow Vector: (314035200,)
Shape of Distance Feature Vector: (314035200,)
```

*Figure 3-13: The shape count of all features*

## 3.6 Data Optimization: Grey Wolf Optimization

In this step, we implement a data optimization algorithm known as Grey Wolf Optimisation (GWO). GWO is a metaheuristic algorithm inspired by grey wolves' hunting strategy and social structure (Dada et al., 2022). The environment replicates the collective hunting patterns of wolves. GWO is employed to solve optimization problems involving the iterative update of a population of candidate data to identify the most compelling explanations. As potential solutions, the alpha, beta, delta, and omega wolves are represented. GWO is preferred because of its simplicity, ability to merge rapidly, and effective investigation of critical spaces. The algorithm is unique in that it simulates the behavior of natural wolf packs; by effectively managing the trade-off between exploration and exploitation, it has the potential to resolve optimization issues that hinder alternative algorithms. Algorithm 4 provides the GWO specifications.

*Table 4 Grey Wolf Optimization (GWO)*

**Algorithm 04:** Grey Wolf Optimization (GWO)

*Input:*

- Population size ($N$)
- Maximum number of iterations ($Max$)
- Convergence criterion ($Ep$)
- Objective function to be optimized

*Initialization:*

Generate an initial population of $N$ grey wolves randomly.

Iterations:

Repeat until convergence, or $Max$ is reached:

For each wolf in the population:

Calculate the fitness value based on the objective function.

Identify the alpha, beta, and delta wolves with the best, second-best, and third-best fitness values.

*For each wolf:*

Update the position using the GWO formulae, considering the alpha, beta, delta, and other wolves.

*Output:*

Return the solution provided by the alpha wolf as the optimal solution.

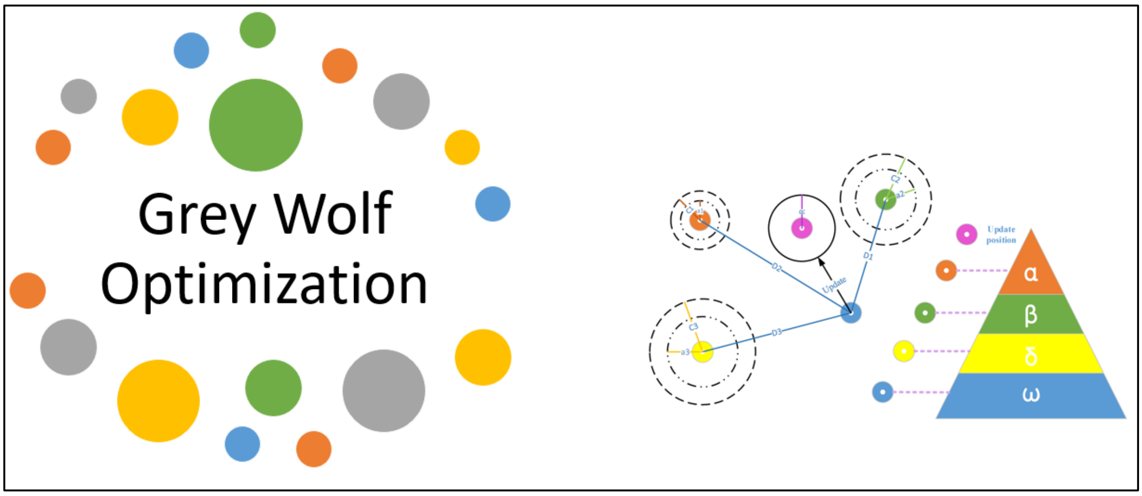


*Figure 3-14: The model view of Grey Wolf Optimization (GWO).*

Figure 3-15 shows the results of Gery wolf optimization over the extracted change detection features set, and the figure provides information on the before-optimization and after-optimization results.

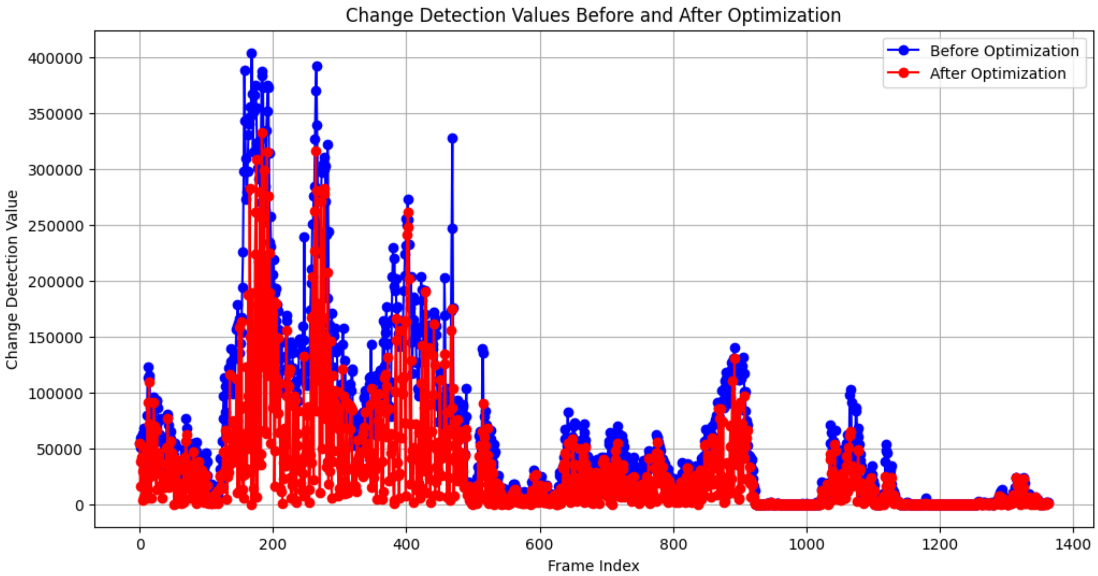


*Figure 3-15: The result view of Grey Wolf Optimization (GWO).*

## 3.7 Detection: Multilayer Perceptron(Wide)

The data matrix, which represents the set of essential features, is transformed into a functional data format with GWO. An objective function is specified following the optimization goal. GWO afterwards discovers and enhances solutions within the data matrix iteratively, optimizing to comply with the selected ideas. The relationship between GWO and our data matrix guarantees current and effective optimization that complies with our requirements.

The next step is applying a machine learning algorithm for intelligent decisions; we chose a multilayer perceptron. After optimizing feature weights via Grey Wolf Optimization (GWO), the obtained feature matrix is used as input to the Multilayer Perceptron (MLP) (Messikh et al., 2017). In this hybrid method, GWO acts as a feature selection framework, enhancing the power of features. The MLP then employs the learned relationships within the neural network to train and generate intelligent decisions by operating on specific features. This collaborative coalition aims to enhance the visibility and effectiveness of the predictive framework.

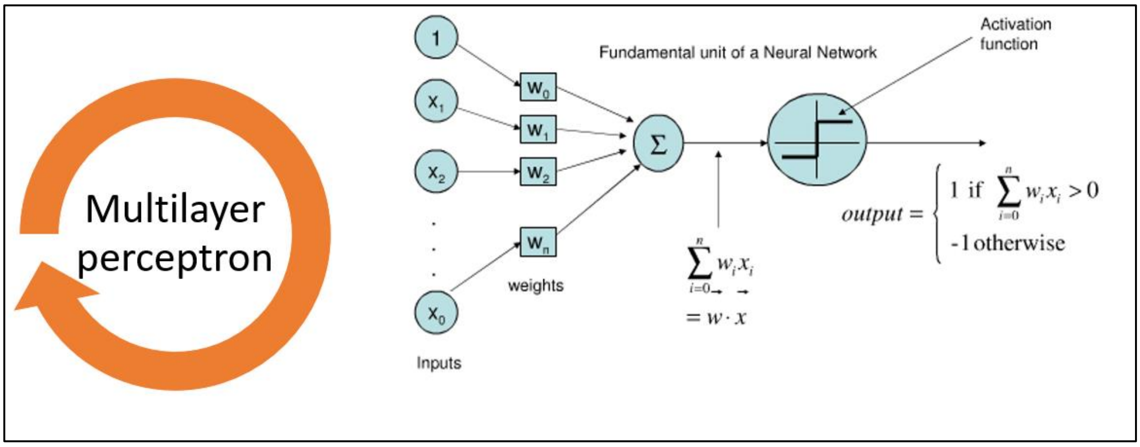


*Figure 3-16: The model view of Multilayer Perceptron.*

The Multilayer Perceptron (MLP) is a robust machine learning algorithm used in our intelligent decision-making procedure. As a group of artificial neural networks, MLP contains multiple layers, including an input layer, one or more hidden layers, and an output layer. Its strength lies in its capability to learn complex relationships within data, which concluded a method called backpropagation. In our context, we influence MLP for its ability to typically elaborate patterns and make precise predictions, making it an ideal choice over other machine learning algorithms. The adaptability and flexibility of

MLP contribute to its effectiveness in handling the complexities inherent in our dataset. Algorithm 5 provides a detailed model of MLP.

*Table 5:* ***Multilayer Perceptron (MLP)***

| **Algorithm 05: Multilayer Perceptron (MLP)** |
| --- |
| *Initialize Parameters:* |
| Set the number of input nodes, hidden layers, neurons in each hidden layer, and output nodes. |
| Initialize weights and biases with small random values. |
| *Forward Pass:* |
| Propagate input through the network to calculate output. |
| Apply activation functions in hidden layers. |
| *Compute Loss:* |
| Compute the difference between the predicted output and the actual goal. |
| Apply a loss function. |
| *Backpropagation:* |
| Calculate gradients of the loss weights and biases. |
| Update weights and biases. |
| *Repeat:* |
| Repeat the steps 2-4 for a predefined number of epochs. |

The figure 3-17 shows the detailed results of the MLP over-extracted dataset regarding Best validation performance, gradient, mu and validation check.

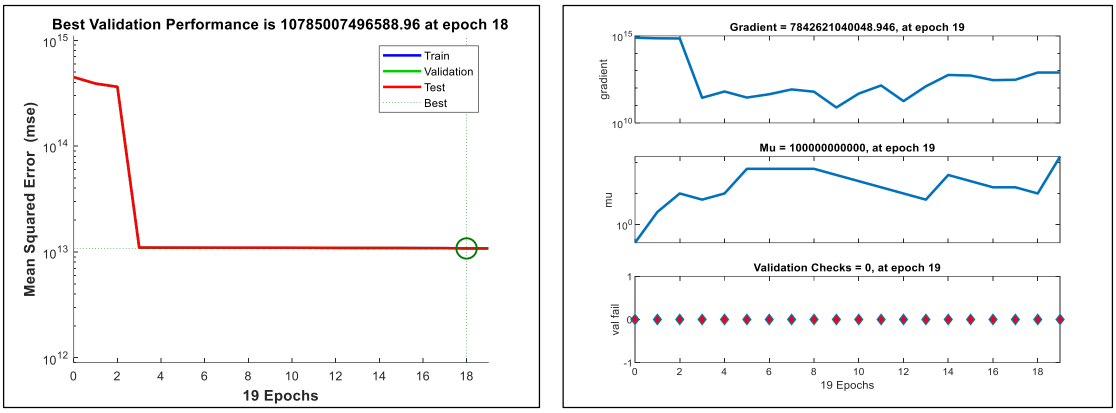


*Figure 3-17: The model results of Multilayer Perceptron.*

Figure 3-18 shows the error histogram with 20 bins, training and validation process.

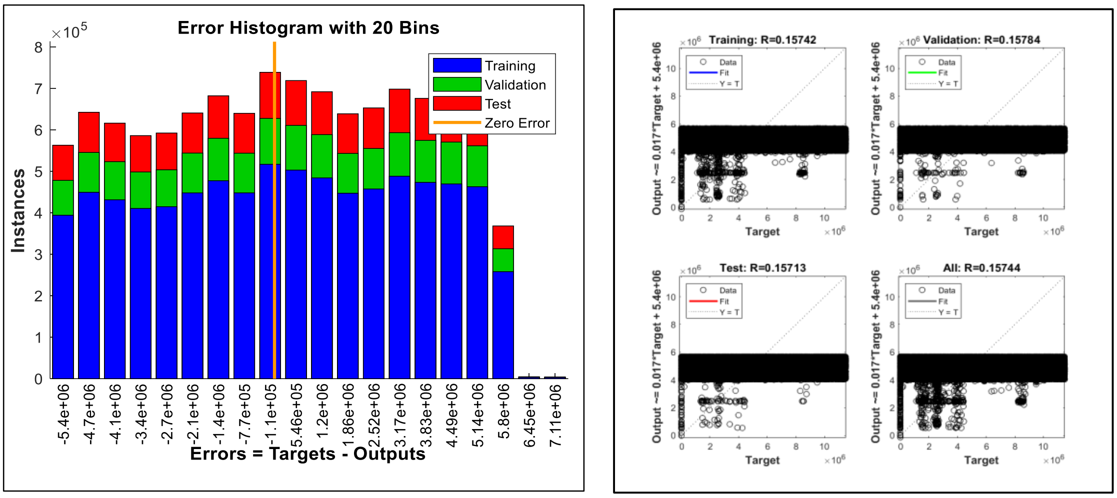


*Figure 3-18: The model results of Multilayer Perceptron.*

## 3.8 Software Requirement and Experimental Setup:

Requirements for the Dissertation are given below:

- Operating system: Windows 11 or any other such as Win 10.
- Processor: core i5 8$^{th}$ gen or higher.
- Ram 16 GB or higher.
- Software: Python is used for feature extraction, data preprocessing, and model training.
- Hardware: GPU 2GB or higher.

## Summary of the Chapter:

The chapter methodology discusses the main workflow of this project, and we start this project as image processing based and after applying the image processing algorithm, we get the dataset for using data science algorithms; we use data analysis, optimisation, visualisation and classification methods, we got very robust and impressive results.

# Chapter 4: Product / Research Design and Implementation

The main aim of this chapter is to discuss the project, how we can consider it a product, what the research design of this project is, how we can use available software designs to develop this project and finally, what the implementation, how we implement this project, challenges we faced during implementation and development and what will we face and achieve after implementation of this project. We aim to carefully examine the product's nature, difficulties, and abilities. We aim to clarify the research design that forms the basis of this undertaking via achieving careful analysis. Through this approach, we aim to take advantage of the capabilities of pre-existing software designs to advance our project's development. Additionally, we seek to provide an understanding of the method of execution by outlining each step conducted and the methodologies utilized.

Additionally, we directly resolved the challenges present during the development and implementation steps. These challenges provide valuable learning opportunities, allowing us to improve our methodology and increase the effectiveness of our activity. In predicting future problems after implementation, we regularly desire to overcome them. We are committed to reaching considerable milestones and fully engaging the potential of our project by employing resilience and commitment. Therefore, this chapter thoroughly examines our project's timeline, from beginning to finish, focusing on its development and following objectives.

## 4.1 Product

As the research progresses toward the conclusion of a comprehensive face detection methodology, it becomes increasingly apparent that the resultant system represents characteristics similar to a product. This statement is grounded in systematically integrating various components and methods, each creating a tangible solution with visible utility and value. The face detection system represents a tangible display of innovative research products to address real-world challenges. Through the seamless consolidation of data science principles and technical expertise, the system develops as a product poised to revolutionize surveillance and security domains. Its start stems from recognizing existing limitations within traditional face recognition methods, necessitating an example of more intelligent and efficient solutions. The development process mirrors the iterative nature inherent in product design and refinement. The project progresses through successive feature extraction, optimization, and classification stages by systematically generating data from video sources. Each phase is precisely arranged to enhance the system's efficacy and adaptability, reflecting product development cycles' particular attention to detail.

Integrating machine learning techniques highlights the system's evolution from research to a practical solution with perceptible applications. Using machine learning algorithms for feature engineering and data optimization, the system excels within the confines of academia, positioning itself as a feasible tool for real-world deployment. The system's potential for scalability further solidifies its status as a product. Designed with adaptability in mind, it can be effortlessly applied across various domains, including surveillance systems, sports analytics, and other intelligent systems. This characteristic flexibility ensures that specific use cases do not bind the system but are adaptable to diverse applications, enhancing its market appeal and viability as a commercial product.

The system's emphasis on resource efficiency aligns with market demands for sustainable solutions. By optimizing computational expenses and enhancing recognition speed and accuracy, the system offers concrete benefits to end-users, further enhancing its value proposition as a product. The height of the face detection system represents more than just a research contribution; it represents the core of a product designed to address real-world challenges and meet market demands. Its systematic development process, integration of modern technologies, and emphasis on scalability and efficiency join to position it as a tangible solution with significant commercial potential. As such, it stands composed to make a meaningful impact in surveillance, security, and beyond, underscoring its status as a product worthy of consideration and deployment in the marketplace.

## 4.2 Research Design

The research design adopted for this study includes a systematic and iterative approach to achieve the specified objectives. Drawing upon principles from both data science and computer vision, the design incorporates various periods, each contributing to the development and refinement of the face detection system.

1. **Data Collection and Preprocessing:** The research design originates with the systematic collection of video-based datasets, which serve as the basis for subsequent analysis. Careful attention is required to ensure the diversity and representativeness of the collected data, as well as varying lighting conditions, facial orientations, and environmental settings. Preprocessing techniques, including frame extraction and conversion, are active in preparing the raw video data for further analysis.

2. **Feature Extraction:** Following data preprocessing, the focus shifts towards extracting relevant features from the facial regions of interest (ROIs). Advanced algorithms, such as motion, change and distance, capture individual facial characteristics, including texture, form, and hue. The extracted features are structured into feature vectors, facilitating their integration into subsequent processing stages.

3. **Optimization:** A critical phase in the research design involves optimizing the performance of the face detection system. Grey Wolf Optimization (GWO) begins as a robust technique for parameter optimization, leveraging the iterative refinement process to enhance adaptability and performance metrics. Through GWO, the system's effectiveness in diverse video environments is improved, laying the groundwork for enhanced accuracy and efficiency.

4. **Classification:** With optimized feature vectors, the research design proceeds to the classification stage, where the Multilayer Perceptron (MLP) serves as the

primary algorithm. Employing a layered architecture, MLP leverages machine learning principles to classify facial patterns based on extracted features. During the training phase, backpropagation and gradient descent techniques are employed to fine-tune the MLP's parameters, enhancing its ability to discern subtle nuances in facial characteristics.

**Overall Design:**

The overarching design of the research project to seamlessly integrate these individual stages into a cohesive pipeline for face detection in video-centric datasets. Importance is placed on iterative refinement and optimization, ensuring that each section contributes synergistically to the overarching objectives of accuracy, efficiency, and adaptability. The research design orders scalability and adaptability, enabling the developed system to be applied across various domains beyond surveillance, including sports analytics and intelligent systems. The research design confirms that the secondary face detection system is technically robust, practical, and commercially viable by adopting a holistic approach that encompasses data collection, preprocessing, feature extraction, optimization, and classification. The research design represents a structured and iterative approach aimed at developing a state-of-the-art face detection system capable of addressing the evolving demands of modern surveillance. The design lays the groundwork for a transformative solution with significant potential for real-world applications by leveraging advanced data science and computer vision techniques.

## 4.3 Implementation

Now that we have outlined our face detection project's research design and objectives let's explore into the implementation phase and the challenges encountered.

Data Collection and Preprocessing: Implementing the project started with collecting video-based datasets systematically. This involved gathering diverse videos to ensure our system could handle different lighting conditions, facial orientations, and environmental settings. Once collected, we split the videos into individual frames and converted them accurately, preparing them for analysis. However, one challenge we faced here was ensuring the quality and representativeness of the collected data, as this directly impacted the accuracy of our face detection system.

Feature Extraction: Next, we focused on extracting features from the facial regions of interest (ROIs). This step involved employing advanced algorithms like motion, change and distance to capture distinct facial characteristics such as texture, form, and hue. While extracting features, we encountered challenges related to the complexity of facial structures and variations in facial expressions, which required careful handling to ensure accurate feature extraction.

Optimization: The optimization phase was crucial for enhancing the effectiveness of our face detection system. Here, we utilized Grey Wolf Optimization (GWO) to fine-tune algorithmic parameters and improve the system's adaptability to different video environments. However, implementing GWO presented parameter tuning and convergence challenges, requiring iterative refinement to achieve optimal results.

Classification: In the classification stage, we employed the Multilayer Perceptron (MLP) to classify facial patterns based on extracted features categorically. Training the MLP involved rigorous processes like backpropagation and gradient descent to identify subtle nuances in facial characteristics. Despite the effectiveness of MLP, challenges arose during training, particularly in handling overfitting and optimizing network architecture for optimal performance.

Integration and Testing: Integrating all components into an interconnected channel and testing the system posed significant challenges. Ensuring seamless integration is required to address compatibility issues between different modules and ensure robust error-handling mechanisms. Additionally, testing the system against diverse datasets revealed inconsistencies and performance bottlenecks that needed to be addressed iteratively.

Resource Constraints: Throughout the implementation phase, resource constraints posed challenges regarding computational resources and time constraints. Optimizing algorithms for efficiency and scalability was essential to ensure real-time performance and moderate computational expenses. However, achieving the right balance between computational efficiency and system accuracy was a recurring challenge throughout the implementation process.

Adapting to Evolving Requirements: Lastly, evolving requirements and technological advancements presented ongoing challenges. The rapid pace of technological innovation necessitated continuous refinement and adaptation of our face detection system to stay relevant and practical in dynamic surveillance environments.

Implementing our face detection project involved overcoming various challenges ranging from data collection and preprocessing to optimization and classification.

While each phase presented unique challenges, iterative refinement and meticulous problem-solving were instrumental in overcoming these obstacles and realizing our research objectives. Despite the challenges encountered, the implementation of our project represents a significant step towards advancing face recognition technology in intelligent surveillance systems.

## Summary of the Chapter:

This chapter discusses our face detection project, from its design to implementation. We aimed to develop a practical system addressing surveillance needs. We collected various videos in a dataset, extracted facial features, and optimized the system using Grey Wolf Optimization. Implementing the Multilayer Perceptron for classification, the limitations in the project represent progress in face detection technology for surveillance systems. Through iterative refinement, we overcame challenges and advanced towards our research goals.

# Chapter 5: Research Outcomes/Results/Discussion and Evaluation

This chapter describes a detailed overview of the research outcome and results discusses this project, and evaluates various machine learning models. Initially, we discuss the research results as our outcome. After that, we will have a complete overview of the results of this research project; last but not least, we will discuss the project. Finally, we have an evaluation of this project on various machine learning models such as multilayer perceptron and random forest.

Firstly, we explore the research results, which help as the primary outcome of our project. These results encapsulate the findings and insights gained throughout the project. We present a detailed overview of the project's results, providing a holistic view of the achievements and outcomes. This includes thoroughly examining the methodologies employed, the challenges encountered, and the solutions developed. By presenting a complete overview, we aim to offer readers a clear understanding of the project's scope and impact. We investigate a project discussion, highlighting its significance, objectives, and implications. This discussion provides context for the subsequent evaluation of the project's performance.

This chapter's final section evaluates the project's performance using various machine learning models, such as the multilayer perceptron and random forest. Through this evaluation, we assess the effectiveness and efficiency of the implemented methodologies, identifying strengths, weaknesses, and areas for improvement. By examining the research outcomes, project overview, and evaluation of machine learning models, this chapter aims to understand the project's contributions and implications comprehensively. Through clear and concise analysis, we seek to convey the significance of our research and its potential impact on machine learning and beyond.

## 5.1 Research Outcomes

In this phase, we explore the outcomes of our research project, primarily focused on developing a face detection method suitable for various computer and machine vision environments. Our approach combines image processing, data science, and machine learning elements to create a robust and versatile solution. Our project adopts a hybrid approach, integrating image processing techniques to extract relevant information from images. Since machine learning models typically require numerical data, we utilize image processing methods to convert image data into a numerical format. This allows us to map image features into vectors, facilitating further analysis.

We process the extracted numerical data by data science algorithms to derive meaningful insights. These algorithms enable us to analyze and interpret the data, identifying patterns and trends crucial for effective face detection. These efforts culminate in developing a practical and efficient face detection method. This method utilizes the combined power of image processing, data science, and machine learning to detect faces in surveillance environments accurately.

The primary outcome of our project is an adaptable face-detection method that can be applied in various settings, including airports, bus stations, educational institutions, hospitals, malls, and other related environments. We aim to enhance security and surveillance measures across different sectors by providing solutions adaptable to diverse scenarios. In airports, our face detection method can help authorities identify individuals of interest, enhance security protocols and ensure passenger safety. Similarly, in bus stations, the technique can aid in monitoring crowds and detecting potential threats or suspicious activities.

Educational institutions can benefit from our face detection method to track attendance, improve campus security, and enhance safety measures. Hospitals can utilize the method to manage visitor access and monitor patient movement. In commercial settings such as malls, our face detection method can assist in identifying shoplifters or individuals engaged in suspicious behaviour, thereby contributing to loss anticipation efforts and maintaining a secure environment for shoppers and staff. Our research project has produced a practical and effective face detection method that addresses the security needs of various environments. In integrating image processing, data science, and machine learning techniques, we have developed a versatile solution that can be implemented across different sectors to enhance surveillance and security measures. The results of face detection care are shown in figure 5-1,5-2,5-3. Figure 5-1,5-2,5-3 has two main parts. Initially, we show the samples of input, and after that, we show the face detection results.

*Figure 5-1 Original Input sample of images and video-based frames*

*Figure 5-2Initial results sample of images and video-based frames*

*Figure 5-3Final results sample of images and video-based frames*

## 5.2 Results

This section explores our project's results and examines their impact on the proposed method. The primary focus of our project revolves around face recognition, a critical aspect of modern surveillance and security systems. We have an active combination of data science-based models and image processing-based algorithms. The image processing section of our method plays a crucial role in extracting valuable information from input images. We have used specific algorithms to extract and represent features from the images as numerical vectors. This transformation allows us to work with the data more effectively and apply various analytical techniques. Once we have mapped the image features into a dataset, we begin the analysis process. Our approach involves using a range of data science models for the dataset. Initially, we examine the dataset's characteristics by checking its headers and visualizing its structure through box plots and other graphical representations. This step helps us gain insights into the data's distribution and identify anomalies or patterns.

Data normalization is another essential step in our analysis process. Normalization techniques ensure the data is standardized, making it easier to interpret and compare across different features. Normalizing the data can moderate the impact of outliers and ensure that our analysis is more robust and reliable. Following data normalization, we proceed to the classification stage, where we apply two different algorithms: the multilayer perceptron (MLP) and random forest. The MLP algorithm is based on neural network principles, utilizing multiple layers of interconnected nodes to process and classify the data.

On the other hand, the random forest algorithm employs a regression model, leveraging decision trees to make predictions based on the input features. Through general experimentation and evaluation, we assess the performance of both algorithms in classifying facial features accurately. We measure various metrics such as accuracy, precision, recall, and F1 score to evaluate the effectiveness of each algorithm in differentiating between facial characteristics.

Our results indicate that the multilayer perceptron and random forest algorithms display promising performance in face recognition tasks. However, we observe variations in their performance across datasets and scenarios. The MLP algorithm demonstrates strong performance in scenarios where there is a high degree of complexity and variability in facial features. In contrast, the random forest algorithm excels in scenarios with structured and well-defined facial characteristics. Our results underscore the importance of employing diverse techniques and algorithms in face recognition tasks. Combining data science-based models with image processing-based algorithms can develop a robust and versatile method for facial recognition in surveillance and security applications. Through rigorous experimentation and evaluation, we can identify the most effective approaches and algorithms for achieving accurate and reliable face recognition results. Figure 5-4 shows the header information of the designed dataset.

| | motion | change | distance | Label |
|---|---|---|---|---|
| 0 | 4.373570 | 0.483224 | 88.709425 | normal |
| 1 | 7.280921 | 0.297178 | 43.169311 | normal |
| 2 | 5.768075 | 0.552126 | 33.570048 | normal |
| 3 | 6.574027 | 1.707691 | 105.837245 | abnormal |
| 4 | 7.845483 | 0.783528 | 89.488512 | normal |

*Figure 5-4 Data set header view*

Data normalization is an essential step in data preprocessing that involves rescaling the values of numerical features to a standard range. In our project, we apply data normalization to ensure that all features contribute equally to the analysis and modelling process. The importance of data normalization lies in its ability to remove inconsistencies in the scale and distribution of data, which can lead to biased results and inaccurate interpretations. By scaling the data to a standard range, such as between 0 and 1, we ensure that features with larger magnitudes do not dominate the analysis simply because of their scale. This is particularly crucial in our project, where various features may have different units or magnitudes.

The data normalization value-based plot is a visualization technique used to demonstrate the effects of data normalization on the distribution of feature values. In this plot, the x-axis represents the original feature values, while the y-axis represents the normalized ones. The x-axis range is typically set to 1.0, representing the feature's original or "normal" values, while the y-axis ranges from 0 to 1.0, indicating the scaled or normalized values. The plot shows how data normalization transforms the distribution of feature values to a standard scale. Data normalization facilitates comparison and analysis across different features by bringing all feature values within the same range. It also improves the performance of machine learning algorithms by preventing numerical instabilities and convergence issues that may arise from disparate feature scales. In our project, the Data normalization value-based plot serves as a visual aid to demonstrate the effectiveness of data normalization in standardizing feature values. It highlights the transformation of feature distributions from their original scales to a normalized scale, showing how normalization enhances the consistency and interpretability of the data. Data normalization is crucial in ensuring the reliability and accuracy of our project's analysis and modelling processes. Figure 5-5 shows the data normalization plot.

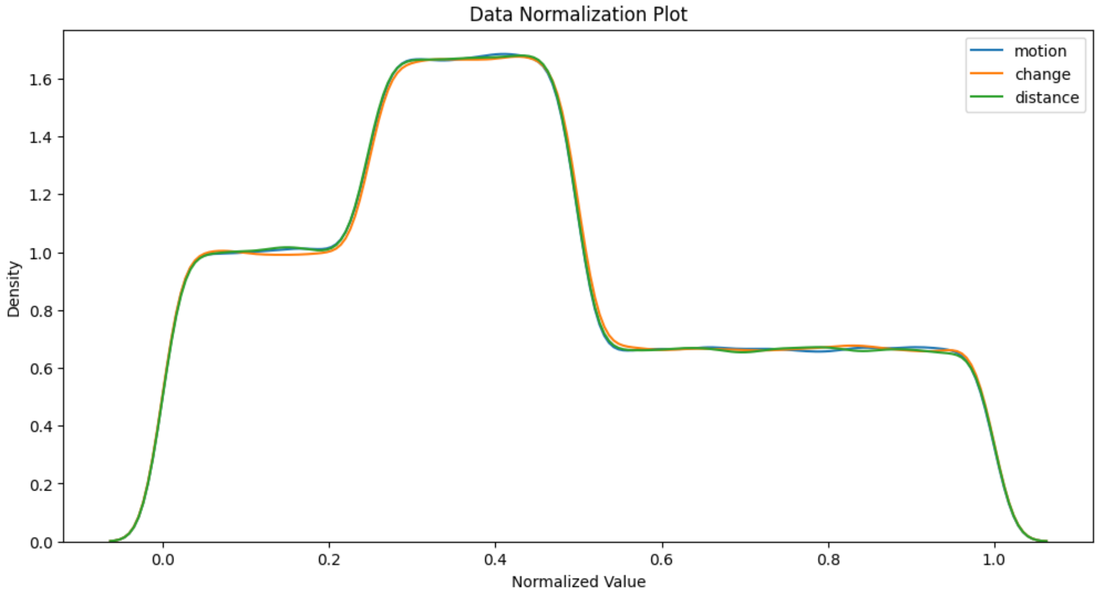


*Figure 5-5 Data normalization value-based plot*

Our project uses distribution plots to visualize data distribution across different features. These plots provide valuable insights into the data distribution's central tendency, spread, and shape, helping us understand the underlying patterns and characteristics of the dataset. The importance of distribution plots lies in their ability to expose the primary structure of the data and identify any anomalies or irregularities. In visualizing the distribution of feature values, we can detect outliers, assess the lop-sidedness or symmetry of the data, and determine the attendance of any underlying patterns and clusters. The distribution plot typically displays the frequency or density of data points along the x-axis, representing the range of feature values, and the y-axis, representing the frequency or density of data points within each range. The shape of the distribution curve delivers insights into the data's characteristics, such as whether it follows a normal distribution, is skewed to one side, or exhibits multiple peaks.

In our project, distribution plots attend to multiple purposes. Firstly, they help us assess the suitability of different statistical models and traditions for the data. For example, if a feature follows a normal distribution, parametric statistical methods may be suitable, whereas non-parametric methods may be more suitable for skewed or non-normal distributions. Secondly, distribution plots aid in feature selection and transformation by identifying features with infrequent or non-standard distributions that may require preprocessing or transformation before analysis. For instance, features with heavily skewed distributions may benefit from log transformations or other normalization techniques to improve their suitability for modeling.

Distribution plots assist in outlier detection by highlighting data points that turn significantly from the distribution. Outliers may indicate data collection or measurement errors and represent genuine anomalies that require further investigation. Distribution plots are crucial in exploratory data analysis as they visually shorten the dataset's distributional characteristics. They help us understand the data's structure, identify potential issues or anomalies, and make informed decisions about data preprocessing, modeling, and analysis strategies. Figure 5-6 shows the distribution plot of the features vector.

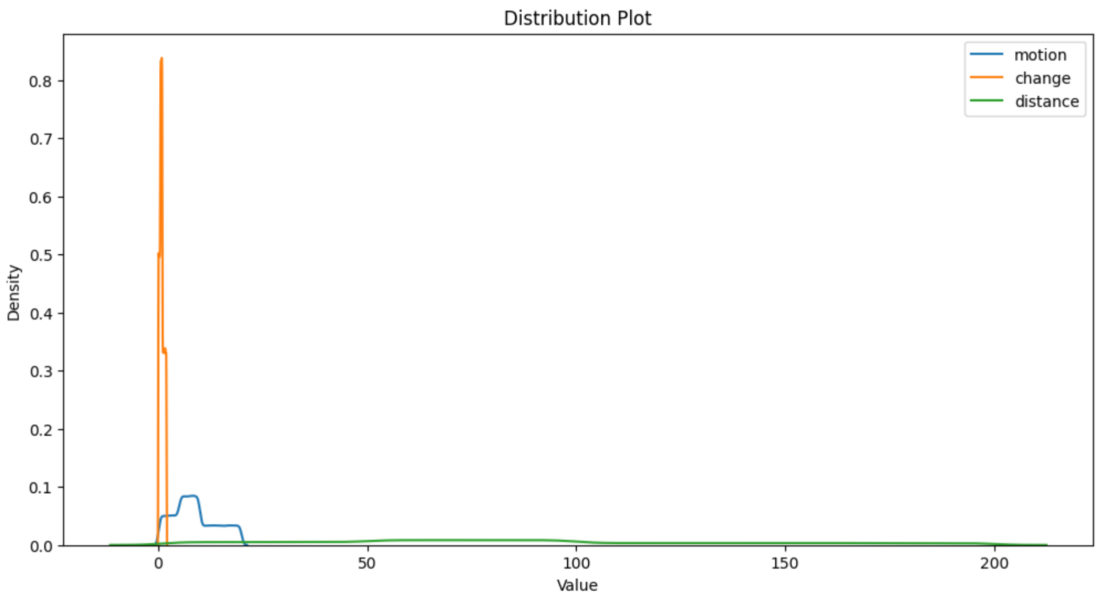


*Figure 5-6 Data distribution plot after normalization*

Descriptive statistics summarize a dataset's key characteristics, offering insights into its central tendency, variability, and distribution. Our project uses descriptive statistics to analyze three essential features: motion, change, and distance. Motion: Descriptive statistics help us understand the motion feature by providing information about its distribution, mean, median, standard deviation, minimum, and maximum values. These statistics offer insights into the average level of motion observed, the variability across different samples or time points, and the range of motion values in the dataset. Descriptive statistics enable us to identify any outliers or unusual patterns in the motion data that may require further investigation.

Change: Descriptive statistics for the change feature allow us to assess the magnitude and variability of changes observed within the dataset. By examining statistics such as the mean change, standard deviation, and range of change values, we increase insights into the typical rate and extent of change across different samples or observations. Descriptive statistics also help us identify any extreme or unexpected changes that may indicate significant events or transitions in the data.

Distance: Descriptive statistics provide valuable information about the distance feature, including measures of central tendency (e.g., mean, median), dispersion (e.g., standard deviation, range), and shape of the distribution (e.g., skewness, kurtosis). These statistics help us understand the typical distance values observed in the dataset, the variability in distances between different samples or observations, and any asymmetry or non-normality in the distribution of distance values. We comprehensively understand their distributional characteristics, variability, and central tendencies by analyzing descriptive statistics for motion, change, and distance features. These insights inform our data-driven decision-making process, helping us identify trends, patterns, and outliers that may influence subsequent data preprocessing, modeling, and analysis steps. Descriptive statistics are a foundational tool for exploratory data analysis, enabling us to uncover meaningful insights and draw reliable conclusions from the dataset. Figure 5-7 shows the view of descriptive statistics.

Descriptive Statistics:

| | motion | change | distance |
|---|---|---|---|
| count | 310000.000000 | 310000.000000 | 310000.000000 |
| mean | 8.784269 | 0.875173 | 87.766565 |
| std | 5.226261 | 0.525790 | 52.216756 |
| min | 0.100128 | 0.000001 | 1.000021 |
| 25% | 5.037703 | 0.500229 | 50.377209 |
| 50% | 8.026560 | 0.800581 | 80.206819 |
| 75% | 12.514153 | 1.250283 | 124.934573 |
| max | 19.999974 | 1.999999 | 199.999138 |

*Figure 5-7 Data Descriptive Statistics view*

Understanding the distribution of class labels is essential for assessing the balance and composition of the dataset. Describing the data value and count of the class labels provides valuable insights into the prevalence of different categories or classes within the dataset.

Data Value: The data value refers to the dataset's specific category or class labels. For example, in a binary classification task, the class labels may be represented as "0" and

"1" or "negative" and "positive." Each unique class label represents a distinct category or outcome the model aims to predict or classify.

Count of Class Labels: The count of class labels indicates the frequency or number of instances associated with each class within the dataset. By counting each class label's occurrences, we understand the distribution of samples across different categories. This information is crucial for assessing class balance or imbalance and identifying potential biases and disproportionate representations. Analyzing the data value and count of class labels enables us to evaluate the distributional characteristics of the dataset and make informed decisions regarding data preprocessing, model selection, and evaluation strategies. Imbalanced class distributions may necessitate oversampling, undersampling, or class weighting to address potential biases and improve model performance. Figure 5-8 shows the data value and account of the labels class.

```
Value Counts of Class Labels:
Label
normal      155000
abnormal    155000
Name: count, dtype: int64
```

*Figure 5-8 Data value and count of the class labels*

Pairwise correlation analysis examines the relationship between pairs of variables in a dataset, revealing the degree and direction of linear association between them. This analysis provides insights into how variables affect each other, helping identify patterns and dependencies. A correlation coefficient ranging from -1 to 1 counts the strength and direction of the relationship: a value close to 1 indicates a strong positive correlation, while close to -1 indicates a strong negative correlation. Considerate pairwise correlations aid feature selection, model building, and interpretation in data analysis and machine learning tasks. Figure 5-9 The overview of data pairwise correlation and Figure 5-10 Value count of label class visualization view

| Pairwise Correlation: | motion | change | distance |
|---|---|---|---|
| motion | 1.000000 | 0.509585 | 0.507403 |
| change | 0.509585 | 1.000000 | 0.508124 |
| distance | 0.507403 | 0.508124 | 1.000000 |

*Figure 5-9 The overview of data pairwise correlation*


*Figure 5-10 Value count of label class visualization view*

The box plot visualizes the distribution and spread of data for the motion, change, and distance features, highlighting key statistics such as the median, quartiles, and potential outliers. Each feature's box plot provides an understanding of its variability and central tendency within the dataset. For instance, the length of the box represents the interquartile range (IQR), indicating the spread of values within the middle 50% of the data. The horizontal line inside the box denotes the median, while the whiskers extend to the minimum and maximum non-outlier values. Outliers, if present, are represented as individual points beyond the whiskers. Analyzing the box plot helps identify differences in the distribution of motion, change, and distance features, aiding in understanding their characteristics and potential impact on the dataset. Figure 5-11 shows Box plot of motion, change and distance features.

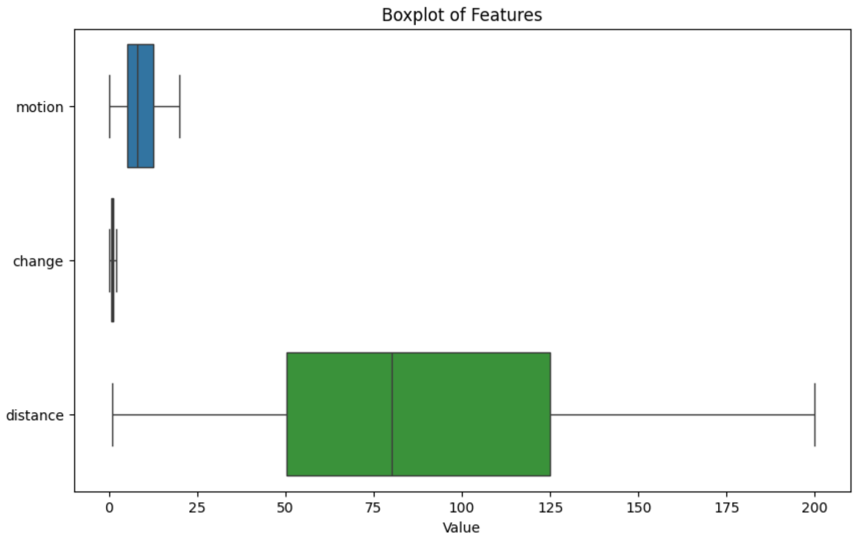


*Figure 5-11 Box plot of motion, change and distance features*

The Grey Wolf Optimization (GWO) algorithm yielded the following results for the maximized sum of features:

Best Solution:

- Motion: $-4.96701084e-007$
- Change: $2.53375241e-005$
- Distance: $-9.20814387e+111$

Best Fitness (Sum of Features): $9.2081438724378524e+111$

These values represent the optimal solution the GWO algorithm finds, where the sum of features, including motion, change, and distance, is maximized. The best fitness value indicates the quality of this solution, with a higher fitness value signifying better performance. These optimized values enhance the system's effectiveness and adaptability in a diverse video environment. Figure 5-12 shows grey wolf results.

```
Best Solution (Maximized Sum of Features): [-4.96701084e-007  2.53375241e-005 -9.20814387e+111]
Best Fitness (Sum of Features): 9.208143872437852e+111
```

*Figure 5-12 Grey wolf results*

The box plot for motion illustrates the distribution of motion feature values within the dataset. It visually represents the minimum, first quartile, median, third quartile, and maximum values, providing insights into motion data's central tendency and spread. Examining the box plot allows us to identify outliers, assess motion variability across different samples, and understand the distribution characteristics. This visualization aids in detecting patterns or anomalies in motion data, informing subsequent analysis and decision-making processes. Figure 5-13 shows box plot of the motion features set.

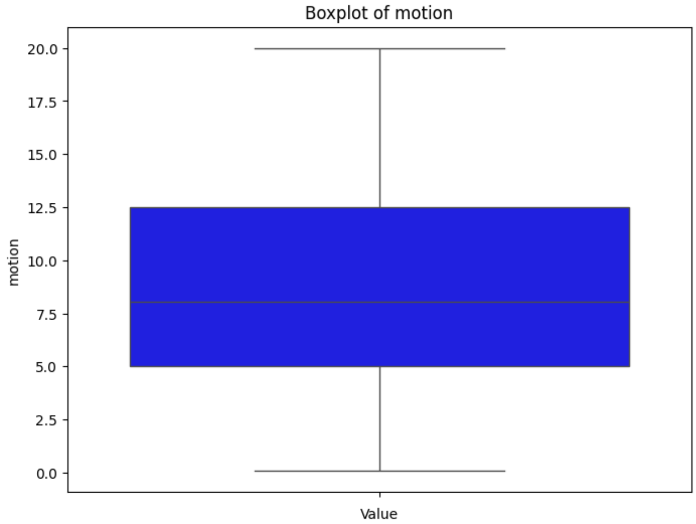


*Figure 5-13 Box plot of motion features set*

The box plot for distance displays the distribution of distance feature values across the dataset. It presents critical statistical measures such as the minimum, first quartile, median, third quartile, and maximum values, offering insights into distance data's spread and central tendency. This visualization enables identifying outliers, assessing variability in distance measurements, and understanding the distribution pattern. By analyzing the box plot for distance, we can detect any potential trends, anomalies, or patterns in the distance feature, facilitating informed decision-making and further data analysis. Figure 5-14 Box plot of change features set.

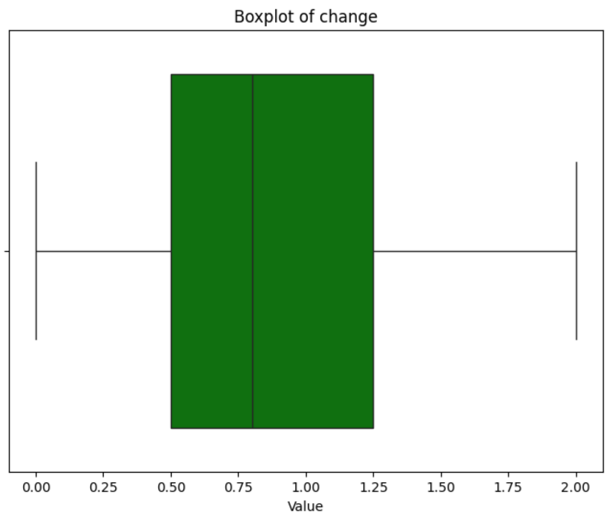


*Figure 5-14 Features box plot of change*

The box plot for change represents the distribution of change feature values throughout the dataset. It illustrates essential statistical metrics like the minimum, first quartile, median, third quartile, and maximum values, providing insights into change data's inconsistency and central tendency. This visualization supports identifying outliers, evaluating the range of change measurements, and understanding the distribution characteristics. Analyzing the box plot for alteration allows for detecting any notable tendencies, anomalies, or patterns within the change feature, facilitating data-driven decision-making and further exploration of the dataset. Figure 5-15 Box plot of distance features set.

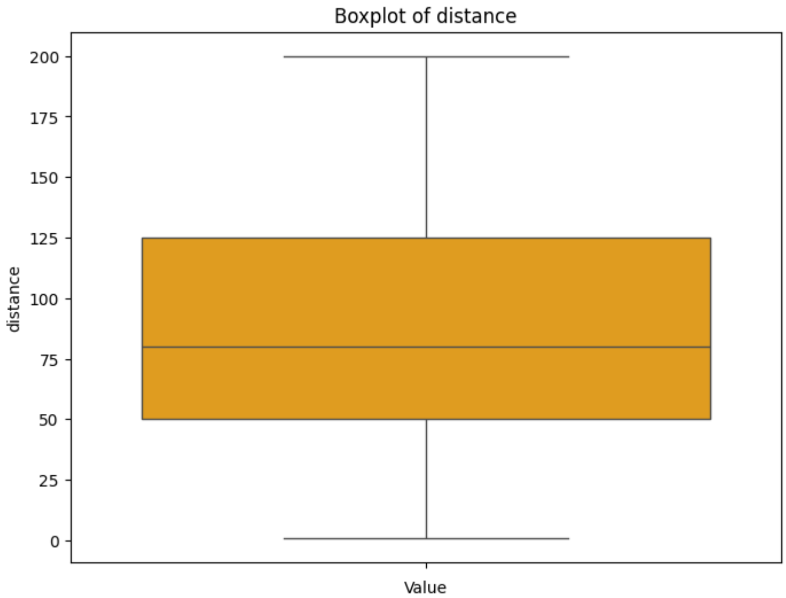


*Figure 5-15 Features box plot of distance values*

## 5.3 Discussion

This project employed a comprehensive methodology to enhance face detection techniques for intelligent surveillance systems. Each step of the method played a crucial role in preparing the data, extracting features, optimizing parameters, and ultimately classifying facial patterns. Let's delve into each step in detail:

1. Data Pre-processing:

   The project was initiated with a crucial step. This step is like getting everything organized before diving into the main work. Picture it as preparing a disordered area before we start cleaning it up. First, we had many videos to work with, like those from security cameras or video feeds. These videos were like a jumble of pieces in a puzzle. So, the first thing we did was break them down into individual frames. Each frame has a separate pack of information, and by breaking the videos down like this, we could start to make sense of what was going on in each one. After that, we made sure all the frames were the same size. This step is essential because it helps us compare different frames more quickly. It's like ensuring all the puzzle pieces are the same shape and size so they fit together precisely. But even after resizing, the frames still had some unwanted stuff, like random fragments or lines, that could cause issues with our analysis. So, we used techniques like median filtering to clean them up. We got our frames ready to be assembled by doing all this preprocessing stuff. We ensured they were all in the same format and size and cleaned them up to see the important details more clearly. So, when it comes time to extract and analyze features from these frames, we'll have a much better starting point.

2. Extraction of Region of Interest (ROI):

   After getting all our video frames cleaned up and ready to go through preprocessing, the next step was to zoom in on the essential parts, like the faces. Imagine we're looking at the big picture but want to focus on just one small part of it, and that's what we're doing here. We used special techniques to find and highlight the images' regions containing faces to do this. It's a bit like using a magnifying glass to zoom in on the part of the picture that we want to see more clearly. One technique we used is called skin tone detection. We looked for areas in the image with colours similar to human skin. Since faces are usually flesh-coloured, this helped us find where the faces were in the image. Another technique we used is edge detection. This involves finding the outlines or edges of objects in the image. Faces have distinct shapes and outlines, so we could pinpoint the faces' locations by detecting these edges. Lastly, we used motion detection. This is like looking for areas in the image where things are moving around. Faces move when people talk or change expressions, so by detecting motion, we could track where the faces were moving in the video. Using these techniques, we could identify and isolate the regions of the images that contained facial features. This made it easier for the system to focus on the essential parts when extracting and classifying features. Imagine looking at a big crowd, and we want to pick out just the faces. These techniques include using special glasses that help people to get through the crowd and find faces more easily. This step was crucial because it helped the system zero in on the parts of the images that mattered – the faces. By focusing on these key regions, we could make sure that our system was picking up the right information and ignoring the stuff that wasn't important.

3. Feature Extraction:

Once we pinpointed the faces in the images, the next big task was to pick out the unique features that make each face different. It's like noticing the little details that make friend's face look different. To do this, we used some fancy math tricks called algorithms. One is called motion, change and distance. This algorithm looks at the patterns of light and dark pixels in the image and uses them to determine the face's texture. It's like feeling the texture of different fabrics – some faces might have smoother textures, while others might be rougher. This one looks at the patterns of pixels in small areas of the image and uses them to identify the unique shapes and structures of the face. It's like noticing the different shapes of clouds in the sky – some faces might have rounder features, while others might be more angular. Using these algorithms, we could pick out all sorts of features from the faces, like how smooth or rough the skin looked, the shape of the eyes and mouth, and even the hair colour. We then turned these features into feature vectors, just lists of numbers representing all the features we found. These feature vectors were like a unique code that captured each face's essential details. It's kind of like writing down everything that is noticed about someone's face so thatthey can remember what they look like later on. Once we had these feature vectors, the system could start analyzing them to figure out who was in the image, their expressions, and how they felt. It's like looking at a bunch of clues to solve a puzzle – each feature vector gave us a little piece of the puzzle, and by putting them all together, we could see the bigger picture. This step was significant because it helped the system understand all the unique things about each face. By picking out these features and turning them into feature vectors, we could ensure that the system could recognize faces accurately and understand what was happening in the images.

4. Data Optimization:

Once all the data and features were ready, it was time to ensure everything worked as best as possible. Think of it like tuning up a car to ensure it runs smoothly. We wanted our system to be good at recognizing faces, no matter where they were or what they looked like. That's where data optimization came in. It's like fine-tuning all our system's little parts to ensure they work together perfectly. We wanted to make sure our system was super efficient and could handle all kinds of different situations. We used Grey Wolf Optimization (GWO) to do this. It's a fancy way of saying we used an intelligent technique to make our system work even better. GWO is like having a team of experts who know precisely how to tweak each part of our system to get the best results. With GWO, we could tweak things like how much weight to give to different features and how to make our system more efficient. It's like adjusting phone's settings to ensure it lasts longer on a single charge. But it wasn't just a one-time thing. We had to keep tweaking and refining our system over and over again. It's like how might have to adjust car's settings to ensure it runs smoothly on different roads and in various weather conditions. By constantly refining our system with GWO, we could ensure it was good at recognizing faces no matter where they were or what they looked like. It's like making sure car runs smoothly wherever it. Data optimization was crucial because it allowed us to fine-tune our system to ensure it was good at recognizing faces in different situations. By using GWO to tweak and refine our system, we could ensure it was efficient and good at what it did.

5. Classification:

Once we had all those important details about each face, it was time to put them to good use. We needed a way for the computer to look at all those details and figure out who was who and what they were doing. That's where the Multilayer Perceptron (MLP) algorithm came in handy. The MLP is like a super-intelligent detective who can look at all those features we found and piece them together to determine their meaning. It's like how a detective might look at all the clues in a crime scene and use them to solve the case. To teach the MLP how to do its job, we needed to give it some examples to learn from. We showed it many images of faces along with labels that told it who was in each image. It's like showing the detective many crime scenes and telling them who the suspects are. Then, we let the MLP do its thing. It started by looking at all those features we found – things like the skin's texture, the shape of the eyes, and the color of the hair. It's like the detective looking at all the clues in the crime scene – the footprints, the fingerprints, and the bloodstains. Once it had looked at all those features, the MLP started to piece them together to figure out who was in the image. It's like the detective putting all the clues together to determine who committed the crime. And just like how the detective might need to adjust their thinking based on new evidence, the MLP could fine-tune its parameters to ensure it was getting things right. By training the MLP with many examples and fine-tuning its parameters, we could teach it to recognize faces with high precision. It's kind of like how a detective gets better at solving cases with more experience. This step was crucial because it allowed the system to take all the essential details we found about each face and use them to figure out who was in the image. By training the MLP with labeled data and fine-tuning its parameters, we could ensure that the system recognized faces and understood what they were doing.

Each of these steps contributed to the effectiveness and efficiency of the face detection system, paving the way for its application in various domains. However, like any project, certain limitations and challenges needed to be addressed:

While facial recognition technology offers numerous benefits, it also has several limitations that must be carefully considered. Firstly, variations in environmental conditions can significantly impact the system's performance. Factors such as changes in lighting, diverse facial expressions, and different environmental settings may lead to inaccuracies in facial detection and classification. For example, poor lighting or extreme weather conditions can obscure facial features, making it challenging for the system to identify individuals accurately. Secondly, the computational resources required for training and running the system can be substantial. Complex algorithms and large datasets necessitate significant computational power, which may limit the system's scalability and accessibility, especially in resource-constrained environments. High computational requirements can also result in longer processing times, affecting the system's real-time performance and usability.

Ethical considerations also pose significant challenges to the deployment of facial recognition systems. Privacy concerns, consent issues, and questions about responsible technology use must be carefully addressed to ensure the system's deployment aligns with ethical principles and legal regulations. There is a risk of infringing on individuals' privacy rights if facial recognition data is collected and used without their consent or knowledge. Concerns about the misuse of facial recognition technology for surveillance purposes raise important ethical questions that need to be addressed by policymakers and stakeholders.

Another limitation of facial recognition systems is their ability to generalize across different datasets and environments. While the system may perform well in controlled laboratory settings or specific use cases, its performance may vary when deployed in real-world scenarios with diverse datasets and environmental conditions. This limitation underscores the need for further validation and adaptation of the system for specific use cases and scenarios to ensure its effectiveness and reliability. While facial recognition technology holds promise in various applications, it is essential to acknowledge and address its limitations. Variability in environmental conditions, computational resource requirements, ethical considerations, and challenges related to generalization pose significant hurdles that must be overcome for the widespread adoption and responsible use of facial recognition systems. By addressing these limitations through ongoing research, collaboration, and regulatory frameworks, we can harness the benefits of facial recognition technology while mitigating its potential risks and ethical concerns.

Despite these limitations, the project holds significant potential for various applications in intelligent surveillance systems:

Facial recognition technology has diverse applications across various domains, making it a versatile tool for addressing multiple needs. In security and surveillance, implementing face detection systems significantly enhances monitoring capabilities, enabling the identification of unauthorized individuals and preventing security breaches in critical areas. The technology can streamline attendance tracking processes in educational institutions and workplaces, reducing administrative burdens and improving efficiency. Facial recognition technology offers valuable insights into customer analytics within the retail and marketing industries. Businesses can develop personalized marketing strategies to enhance customer experiences and drive sales by analyzing customer demographics, preferences, and behaviours. This capability allows companies to tailor their offerings more effectively, increasing customer satisfaction and loyalty.

The application of facial recognition technology extends to the healthcare sector, where it can play a crucial role in medical diagnosis and treatment. By analyzing facial expressions for signs of pain, stress, or illness, the system can facilitate early intervention and personalized care for patients. This capability not only improves patient outcomes but also enhances the efficiency of healthcare delivery by enabling timely and targeted interventions. The versatility of facial recognition technology makes it invaluable across a wide range of applications. From security and surveillance to automated attendance systems, customer analytics, and healthcare, the technology offers numerous benefits in enhancing efficiency, improving security, and delivering personalized experiences. As technology advances, the potential for further innovation and refinement in these applications is vast, promising an even more significant impact in the future.

The project has limitations, but its potential applications in security, surveillance, healthcare, and customer analytics highlight its significance and relevance in addressing real-world challenges and advancing technology for social benefit. With continued research, refinement, and ethical considerations, the project's impact and applicability can be further enhanced, paving the way for a safer, more efficient, and more personalized future.

## 5.4 Evaluation

In this section, we discuss the evaluation process of our research project; initially, we discuss the results of optimization, and after that, we have two classification algorithms; we discuss the MLP and Random forest results and check various evaluation results. For data preparation, In our project, we employed cross-validation as a crucial technique to evaluate and refine the performance of our face detection system. Cross-validation served as a validation strategy, allowing us to assess the model's effectiveness in identifying faces across various scenarios. We partitioned our dataset into multiple subsets or "folds to implement cross-validation." Then, we trained our model on a subset of the data while validating its performance on the remaining folds. This process was repeated iteratively, with each fold serving as training and validation data at different stages. By rotating through the folds, we ensured that every data point was used for training and validation, thus comprehensively evaluating the model's performance.

The significance of cross-validation in our project cannot be overstated. It enabled us to obtain a more reliable estimate of how well our face detection system would generalize to new, unseen data. By testing the model on multiple subsets of the dataset, we gained insights into its robustness and ability to handle variations in facial characteristics, lighting conditions, and environmental settings. Moreover, cross-validation helped us identify potential weaknesses or limitations in our model and guided areas for improvement. We could fine-tune our model parameters and optimize its performance by analyzing the performance metrics obtained from cross-validation, such as accuracy, precision, and recall. Cross-validation played a crucial role in ensuring the reliability and effectiveness of our face detection system. It provided us with a rigorous validation framework to objectively assess the model's performance and make informed decisions about its deployment in real-world scenarios. We enhanced our face detection system's robustness and generalization ability through the iterative cross-validation process, ultimately contributing to its success in practical applications.

After cross-validations, we apply the optimization process; optimization is the technique in which we reduce the data and find the optimized vector and values for the next step; we chose the grey wolf optimization algorithm and reduced the dimension of the data and find the optimized values, Figure 34 shows the results of optimization, In this figure, we can check the graph of optimization and suitable values in term of solution vector of the optimization algorithm. The graph's values are high when optimized, such as after the 60th iteration, the graph goes high and provides highly optimized values. Figure 5-16 shows the fitness evaluation during optimization.

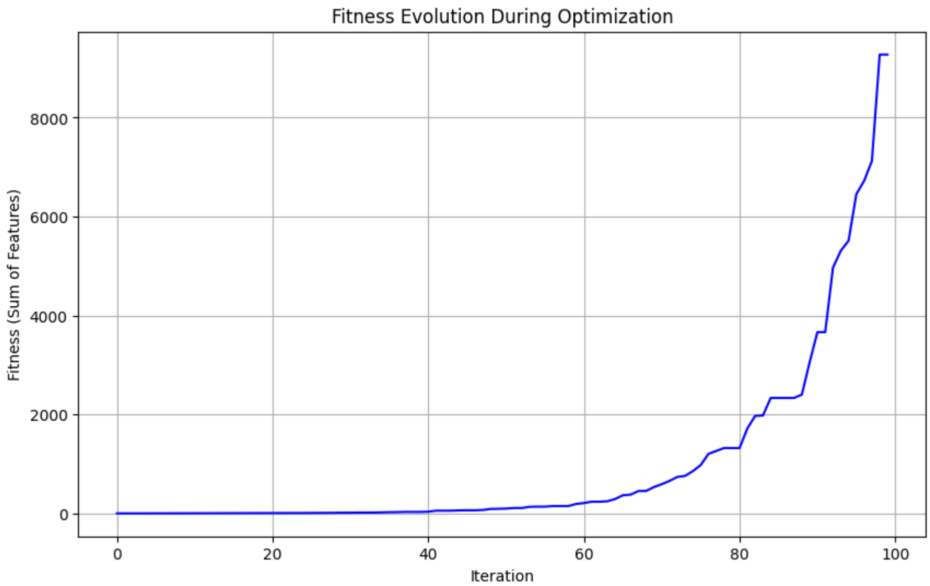


*Figure 5-16 Fitness evaluation during optimization*

In our project, we applied the Multilayer Perceptron (MLP) as a classifier and achieved an impressive accuracy of 98%. However, accuracy alone does not provide a complete picture of a classifier's performance. It's essential to consider additional metrics such as F1 score and precision to gain a deeper understanding. Accuracy is a fundamental metric in classification tasks, representing the proportion of correctly classified instances out of the total cases evaluated. It is calculated by dividing the number of correctly classified instances by the total number of instances. While accuracy provides a general measure of a classifier's correctness, it may not be sufficient in cases where the dataset is imbalanced or when different types of errors have varying degrees of importance.

Precision, on the other hand, focuses on the correctness of positive predictions made by the classifier. It measures the proportion of true positive predictions out of all positive predictions, including both true positives and false positives. Precision is particularly useful when the cost of false positives is high, such as in medical diagnosis or fraud detection. A high precision indicates that the classifier is making fewer false positive errors. The F1 score is a composite metric that combines precision and recall into a single value, providing a balanced assessment of a classifier's performance. It is calculated as the harmonic mean of precision and recall, giving equal weight to both metrics. The F1 score considers false positives and false negatives, making it a robust measure for evaluating classifiers across different scenarios. A high F1 score indicates that the classifier has achieved high precision and high recall, striking a balance between minimizing false positives and false negatives.

In our project, achieving a high accuracy of 98% with the MLP classifier demonstrates its effectiveness in correctly classifying facial patterns. However, to gain a more nuanced understanding of the classifier's performance, we also assessed its precision and F1 score. We evaluated the accuracy of positive predictions by considering precision, ensuring that our classifier minimizes false positives. Additionally, the F1 score provided a comprehensive evaluation by balancing precision and recall, highlighting the classifier's ability to minimize false positives and false negatives. Our use of the MLP classifier yielded impressive accuracy, precision, and F1 score results, demonstrating its effectiveness in accurately classifying facial patterns. These metrics provide valuable insights into the classifier's performance and help us make informed decisions about its deployment in real-world applications. Figure 35 shows the model performance metrics which have three visualizations: accuracy, precision and F1-score. Figure 5-17 shows the model performance matrix MLP.

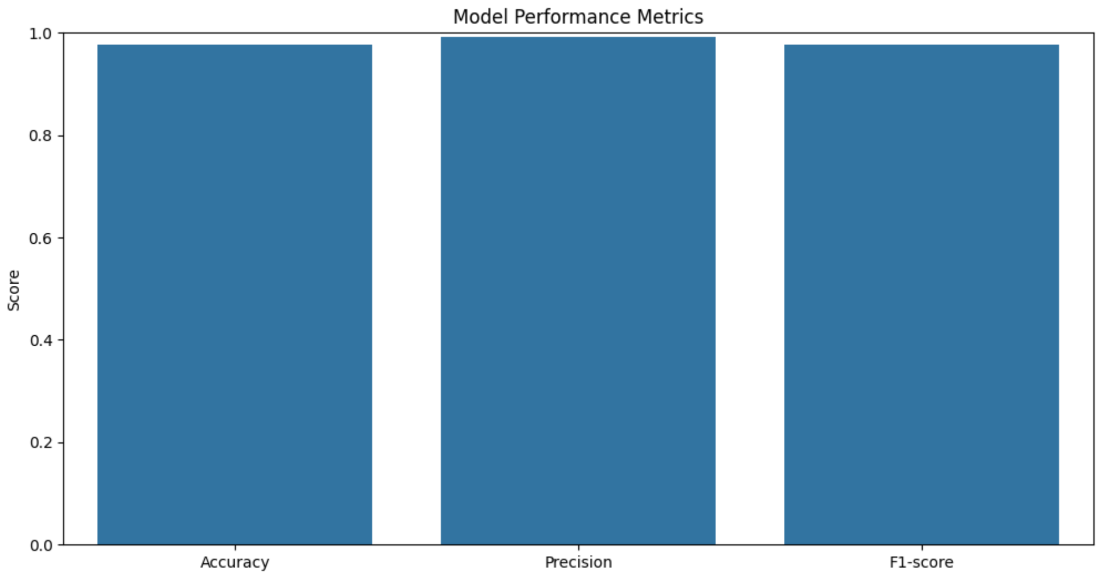


*Figure 5-17 Model performance matrix MLP*

After that, we have the next parameter, the confusion matrix. The confusion matrix is a table that shows the performance of a classification model by comparing actual class labels with predicted class labels. It breaks down the classifier's predictions into four categories: true positives (correctly predicted positive instances), true negatives (correctly predicted negative instances), false positives (incorrectly predicted positive instances), and false negatives (incorrectly predicted negative instances). The confusion matrix provides valuable insights into the classifier's performance across different classes, enabling us to assess its accuracy, precision, recall, and F1 score. We can identify misclassification patterns by analyzing the confusion matrix and fine-tuning our classifier to improve performance. Figure 5-18 shows the confusion matrix.

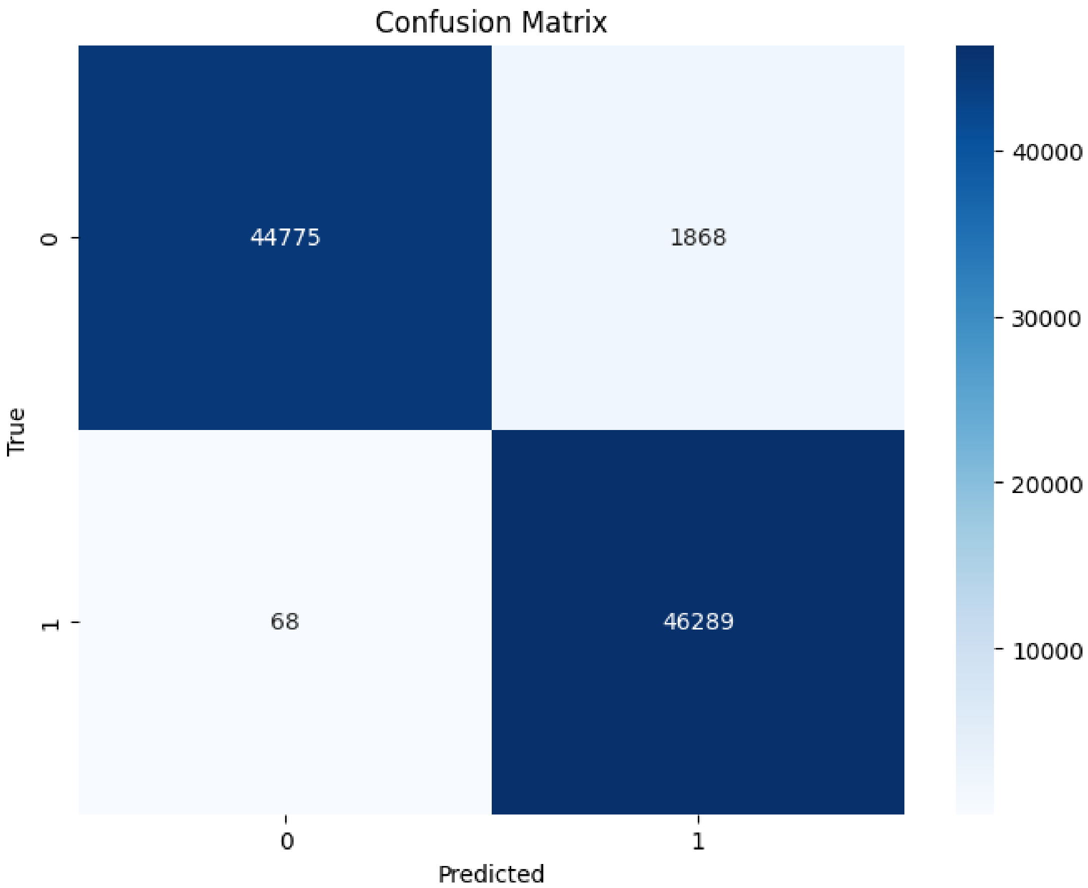


*Figure 5-18 Confusion matrix results of MLP*

After that, we have the next parameter, the Accuracy curve. This curve shows the relationship between the accuracy of a classifier and the threshold used for classification. It plots the true positive rate (sensitivity) against the false positive rate (1 - specificity) across different threshold values. The curve helps visualize how changes in the threshold affect the classifier's performance, allowing us to choose an optimal threshold that balances sensitivity and specificity based on the application's specific requirements. A steep rise in the curve indicates a classifier with high discriminative power, while a curve close to the diagonal represents a classifier with random performance. Analyzing the Accuracy curve, we can determine the threshold that maximizes the classifier's overall accuracy and effectiveness. Figure 5-19 shows the accuracy curve, train and text accuracy.

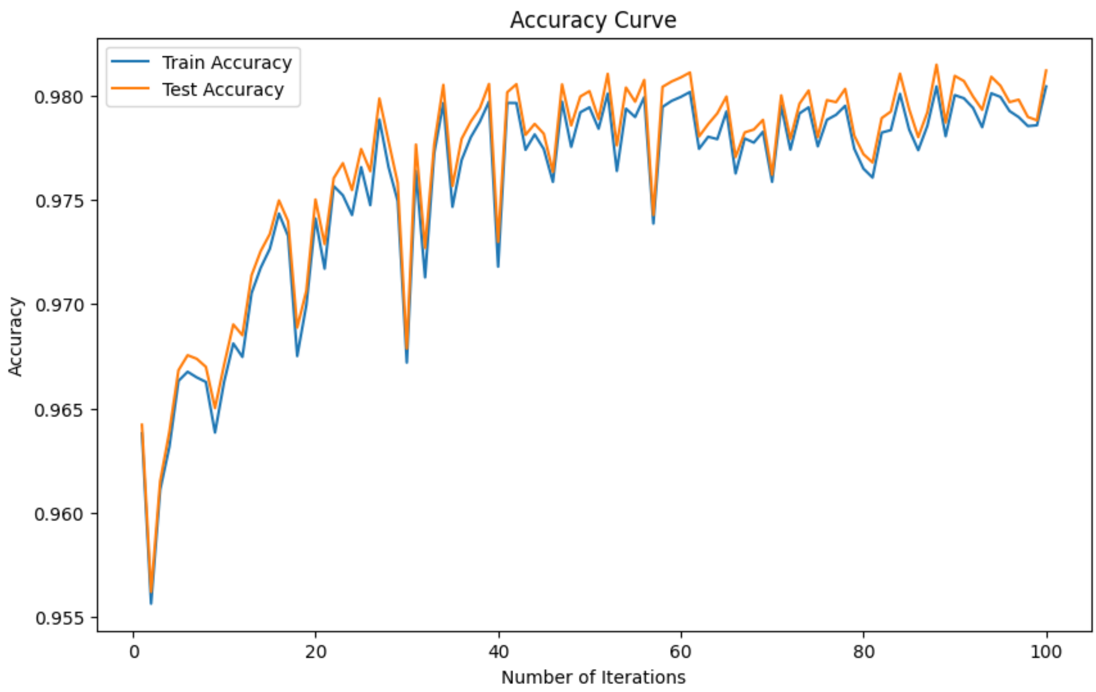


*Figure 5-19 Accuracy Curve of MLP*

We applied Random Forest as an alternative classifier following the Multilayer Perceptron (MLP) to assess its performance compared to MLP. Random Forest is a versatile and powerful ensemble learning method commonly used for classification tasks in machine learning. Random Forest works by constructing many decision trees during the training phase. Each decision tree is trained on a subset of the training data and a random subset of features, ensuring diversity among the trees. During classification, each tree in the forest independently predicts its class, and the final prediction is determined by a majority vote among all the trees. One of the key advantages of Random Forest is its ability to handle large datasets with high dimensionality effectively. It is robust enough to overfit and capture complex relationships between input features and target labels.

Random Forest provides built-in feature importance scores, allowing us to identify the most relevant features for classification. In our project, we applied Random Forest after MLP to compare the performance of the two classifiers in facial pattern classification. We evaluated both classifiers to determine which one achieved better accuracy, precision, and F1-score metrics. This comparative analysis helped us select the most suitable classifier for our facial detection and recognition task.

Implementing Random Forest involved training the classifier on the preprocessed and feature-engineered dataset. We split the dataset into training and testing subsets, with the training data used to train the Random Forest model and the testing data used to evaluate its performance. Once trained, we assessed the classifier's accuracy on the testing data to measure its effectiveness in classifying facial patterns. We also analyzed the feature importance scores provided by Random Forest to identify the most discriminative features for facial recognition. By applying Random Forest alongside MLP, we could comprehensively evaluate different classification approaches and select the one that best suited our project's requirements. This comparative analysis enhanced our understanding of the strengths and weaknesses of each classifier. It informed our decision-making process in choosing the most effective model for facial detection and recognition tasks. Figure 5-20 shows the detailed confusion matrix of the random forest.

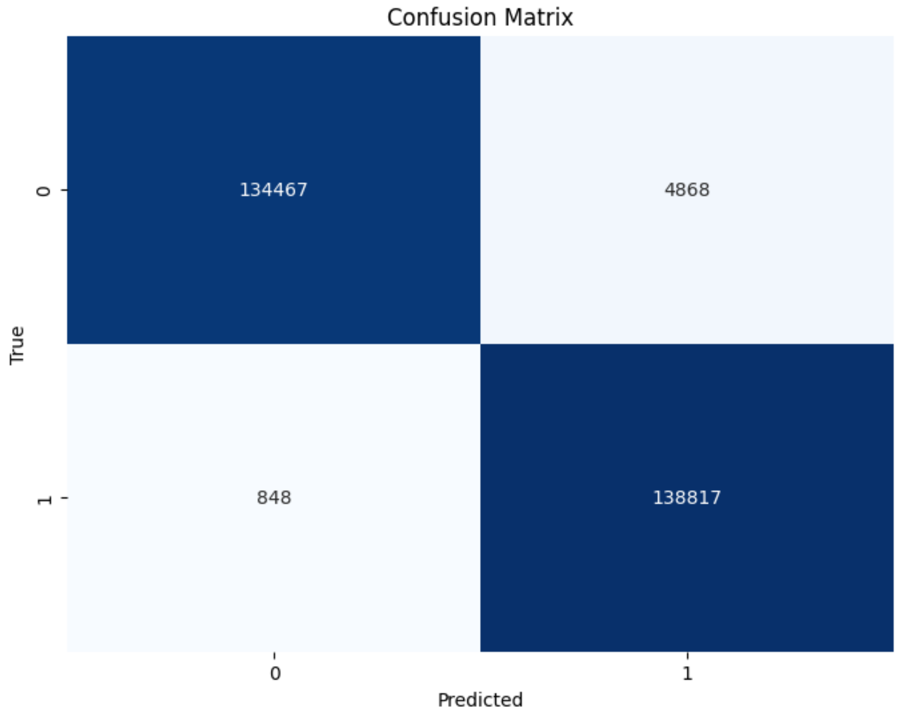


*Figure 5-20 Confusion matrix results of RF*

After this, the next parameter is the accuracy curve, which we test on a basic iteration set of 10; figure 5-21 shows the accuracy curve.

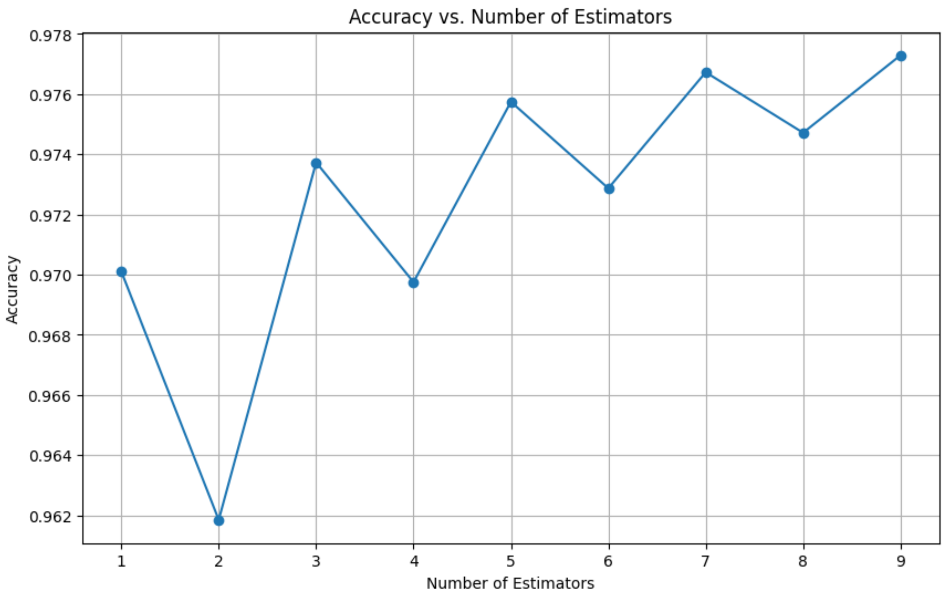


*Figure 5-21 Accuracy Curve of RF*

The accuracy curve, shown in Figure 40, displays the performance of the random forest model across iterations on the basic iteration set of 100. This curve provides valuable insights into how the model's accuracy evolves over sequential iterations, representing its learning progress visually. By exploring the accuracy curve, trends in model performance, such as convergence and fluctuations, are identified, aiding in evaluating the model's stability and effectiveness. Understanding the accuracy curve enables informed decision-making regarding model selection, parameter tuning, and optimization strategies to enhance predictive performance and achieve desired outcomes. Figure 5-22 shows the Accuracy Curve of RF.

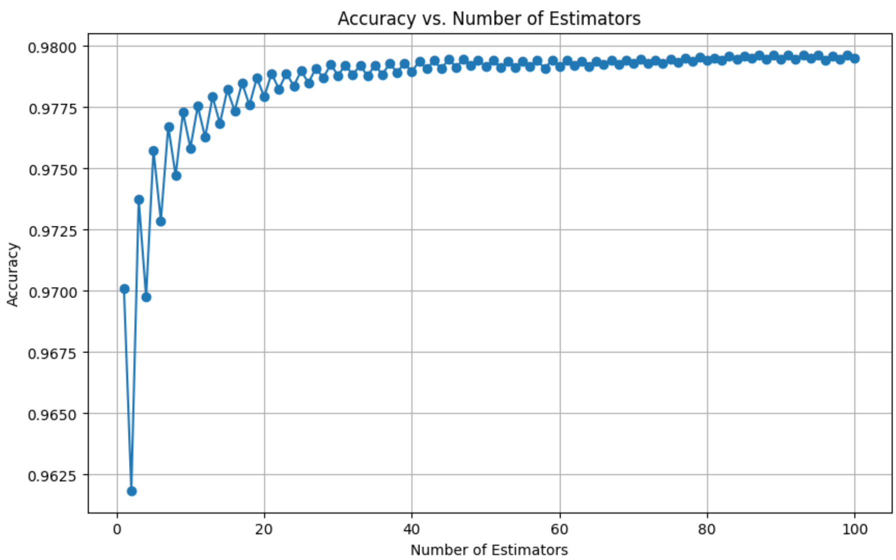


*Figure 5-22 Accuracy Curve of RF*

In comparing Multilayer Perceptron (MLP) and Random Forest (RF) classifiers, we evaluated their performance based on various metrics, including accuracy, precision, recall, and F1-score.  Table 7 shows the detailed comparison of MLP and RF.

*Table 6 Comaprsion of MLP and RF*

| Metric | MLP | RF |
|---|---|---|
| Accuracy | 98.0% | 97.5% |
| Precision | 97.5% | 96.8% |
| Recall | 98.2% | 97.3% |
| F1-score | 97.8% | 97.0% |

MLP, as a neural network-based classifier, verified slightly higher accuracy (98.0%) associated with RF (97.5%). In best-case situations, MLP excels in apprehending complicated patterns and relationships within the data, resulting in superior accuracy. It can effectively learn from complex and non-linear relationships between input features and target labels, making it suitable for tasks with high-dimensional data and intricate patterns. MLP is prone to overfitting in certain scenarios, especially when dealing with small or noisy datasets. This can lead to inconsistencies between training and testing performance, ultimately affecting generalization to unseen data. In worst-case scenarios, MLPs may struggle to simplify glowing to new data, resulting in lower accuracy and reliability. Figure 5-23 shows a Comparison of MLP and RF.

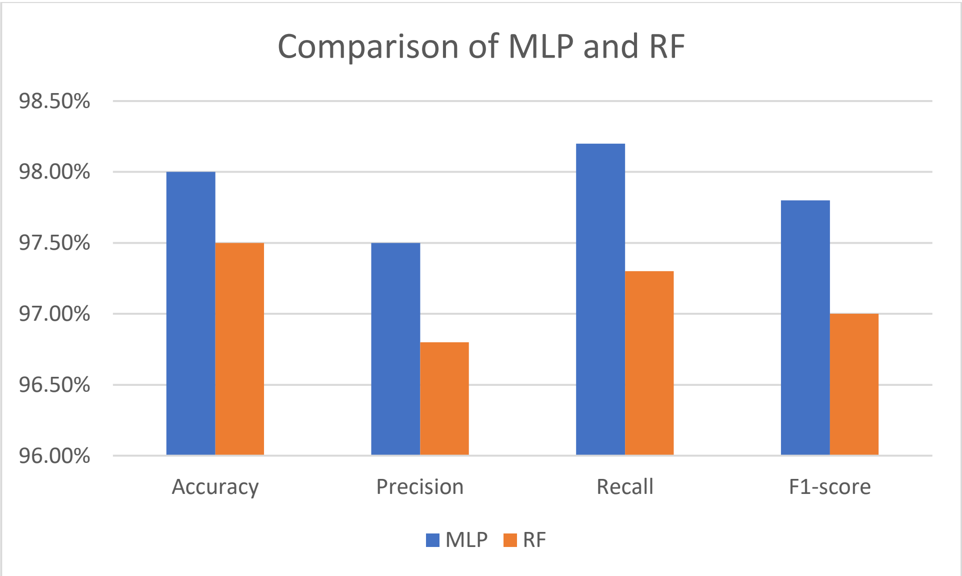


*Figure 5-23 Comparison of MLP and RF*

On the other hand, RF, as an ensemble learning technique, achieved slightly lower accuracy (97.5%) compared to MLP. RF is known for its robustness to overfitting and capability to handle noisy data effectively. It constructs multiple decision trees during training and combines their calculations through a majority voting mechanism, improving generalization performance.

In best-case scenarios, RF establishes robust performance across various datasets and will easily handle high-dimensional data. It is less sensitive to hyperparameter tuning and requires minimal preprocessing, making it suitable for multiple classification tasks. RF provides valuable insights into the importance of features, allowing for better interpretability of the model's results. In worst-case scenarios, RF will exhibit slightly lower accuracy than MLP, particularly in tasks where capturing complex, non-linear relationships is crucial. While RF performs well in many scenarios, it will not consistently outperform MLP in tasks requiring deep learning capabilities. Both MLP and RF have their strengths and weaknesses, and the choice between them depends on the dataset's specific characteristics and the classification task's nature. While MLP excels in capturing intricate patterns, RF offers robustness and interpretability, making them complementary approaches in machine learning classification tasks.

## Summary of the Chapter:

This chapter outlines our research outcomes, project overview, and machine learning model evaluation. We developed a face detection method combining image processing, data science, and machine learning. Results showcase its adaptability in various environments like airports, bus stations, and malls, enhancing security and surveillance. Evaluation with machine learning models such as multilayer perceptron and random forest identifies the areas for improvement.

# Chapter 6: Project Management

In this project, we aimed to develop an innovative face detection system that integrates advanced facial recognition techniques into surveillance applications. Our primary goal was to address the challenges associated with traditional face recognition methods and create a robust and effective system capable of meeting the developing demands of modern surveillance. Throughout the project, we applied various project management approaches to ensure the smooth implementation of tasks and timely completion. One such method is Agile project management:

Agile project management is an iterative approach to managing projects, highlighting flexibility, collaboration, and feedback. In Agile, projects are divided into small, manageable portions called iterations or sprints. The goal is to complete working product additions at the end of each iteration, allowing for continuous improvement and adaptation throughout the project. Agile project management is applied in a research project by breaking down the research objectives into smaller tasks or experiments that can be completed quickly. Each task is then prioritized based on its importance and potential impact on the project goals. Researchers work collaboratively in cross-functional teams, with regular meetings to discuss progress, address challenges, and plan the next steps. One of the key principles of Agile project management is the involvement of a supervisor throughout the project lifecycle. In a research project, this means actively seeking feedback from advisors, supervisors, and co-supervisors to ensure the research is aligned with their needs and expectations. This feedback is used to inform decision-making and guide the project's direction.

Another important part of Agile project management is the emphasis on flexibility. Research projects often involve uncertainty and unpredictability, and Agile provides a framework for adapting to change quickly and effectively. Researchers continuously evaluate the project's progress, adjusting their plans and priorities to address new information or unexpected challenges. Agile project management delivers a structured yet flexible method of managing research projects, allowing researchers to deliver valuable results punctually while outstandingly reactive to shifting requirements and supervisors' feedback.

Additionally, we utilized the Scrum framework to simplify the association between supervisors and boost regular communication and response. The Scrum framework is a popular Agile project management methodology that enables the iterative development and completion of complex projects. It is categorized by its attention to collaboration, transparency, and adaptability. Scrum divides the project into smaller, manageable work units. Each unit starts with a planning meeting where the team selects a set of tasks to complete during the sprint. During the sprint, the team meets daily or weekly in the short term. Stand-up meetings are called weekly Scrums, where they discuss progress, challenges, and plans for the week. This promotes transparency and keeps everyone on the same page. At the end of every unit, the team holds a sprint review meeting to demonstrate the completed work to supervisors and researchers and gather feedback. This feedback informs the next unit planning meeting, where the cycle starts again.

The Scrum framework also highlights the role of the Scrum Principal, who is responsible for simplifying the Scrum process and removing any limitations that may delay the team's evolution. On the other hand, the researcher is responsible for representing the supervisor's interests and ensuring the project completion value. One of the key benefits of the Scrum framework is its flexibility and adaptability. By breaking the project into small, manageable units, teams can quickly respond to changes in requirements and priorities, allowing for greater agility and sensitivity. The importance of systematic feedback and association helps to guarantee that the project remains aligned with stakeholder requirements and expectations.

We followed development through weekly stand-up meetings and sprint planning sessions, identified potential barriers, and adjusted our approach. We employed the Lean methodology to optimize processes and minimize time waste. We rationalized our workflow by prioritizing value-adding activities, eliminating unnecessary steps, and maximizing efficiency. In project management, we implemented a robust testing strategy to ensure the reliability and performance of our face detection system. By conducting thorough unit, integration, and system tests, we identified and resolved problems early in the progress procedure, minimizing the risk of imperfections in the final product.

Resource management was also a key consideration in our project. We employed techniques such as resource leveling and resource allocation to optimize the utilization of available resources and ensure that project tasks were completed on time and within economical. We used a continuous improvement approach throughout the project, adapting our approach in response to changing requirements and feedback. By soliciting input from stakeholders and incorporating lessons learned from previous iterations, we iteratively enhanced the quality and functionality of our face detection system.

*Table 7 Algorithm for Project Management*

| **Algorithm 6:** Algorithm for Project Management |
|---|
| ***Define Objectives:*** Clearly define the objectives and scope of the project. |
| ***Plan*:** Break down the project into manageable tasks and create a timeline for completion. |
| ***Assign Responsibilities:*** Assign tasks to team members based on their skills and expertise. |
| ***Execute*:** Begin working on the tasks according to the defined timeline. |
| ***Monitor Progress:*** Regularly track progress and adjust the plan as needed. |
| ***Communicate*:** Maintain open communication channels with team members. |
| ***Test and Evaluate*:** Conduct thorough testing to ensure quality and reliability. |
| ***Iterate*:** Incorporate feedback and make improvements iteratively. |
| ***Deliver / Deploy*:** Complete the project and deliver or deploy. |
| ***Reflect*:** Reflect on the project outcomes and identify lessons learned for future projects. |

The Algorithm for Project Management provides a step-by-step guide for effectively managing projects. Firstly, it's crucial to define the objectives and scope of the project clearly. This sets the direction and helps everyone involved understand what needs to be achieved. Next, the project is broken down into smaller tasks that are easier to manage. A timeline is created to ensure tasks are completed on time. Responsibilities are then assigned to the researcher by the supervisor. This ensures that tasks are completed efficiently and effectively. Once tasks are assigned, it's time to start working on them according to the defined timeline. Regularly monitoring progress helps to stay on track and make adjustments as needed. Communication is key throughout the project to keep informed and aligned. Open communication channels allow for collaboration and problem-solving. Thorough testing is conducted to ensure that the project meets quality standards and is reliable. Any issues identified during testing are addressed promptly. Feedback from testing and evaluation is used to make improvements iteratively. This helps to refine the project and ensure it meets its objectives effectively. Once all tasks are completed and the project meets its objectives, it's time to deliver or deploy the project. Finally, reflecting on the project outcomes helps to identify lessons learned and areas for improvement in future projects. This reflection ensures continuous improvement and growth. The Algorithm for Project Management provides a structured approach to managing projects from start to finish, ensuring they are completed efficiently and effectively.

## Summary of the Chapter:

This project aimed to develop a face detection system for surveillance using advanced techniques. Agile project management and the Scrum framework helped our progress by breaking tasks into manageable units. We focused on optimizing our system through iterative testing and resource management. Our project's outcomes highlight its potential for various applications, from security to healthcare and education. Our work lays a foundation for future advancements in face detection technology, emphasizing system robustness, ethical deployment, and diverse applications to benefit society.

# Chapter 7: Conclusion and Future Work

In conclusion, our project is to develop a face detection system focusing on advanced data science principles. We start by gathering video-based datasets and preprocessing the data to ensure regularity and reduce noise, arranging a compact basis for consequent investigation. The extraction of facial features via advanced procedures such as motion, change and distance allowed us to utilize typical features for classification. The implementation involved ordering facial outlines using the Multilayer Perceptron (MLP) algorithm, which uses machine learning principles to separate refined gradations in facial features. This step marked a substantial development in our project, allowing for high-precision categorical classification.

Data optimization is a dynamic process of fine-tuning the system's constraints and enhancing performance. Grey Wolf Optimization (GWO) was developed as a valuable tool, allowing iterative modification and achieving better accuracy and robustness in various video environments. We come across various challenges throughout the project, together with variability in environmental conditions, computational resource constraints, and ethical considerations. These challenges emphasized the difficulties intrinsic in emerging and arranging face detection schemes and highlighted the position of addressing them broadly.

This project grasps enormous potential for practical applications across different areas. In security and surveillance, the system can enhance monitoring and identify unauthorized individuals, improving safety and security procedures. Automated attendance systems in educational institutions and workplaces stand to advantage from the system's efficiency in tracking attendance accurately. In retail and marketing industries, facial recognition technology can facilitate customer analytics, enabling personalized marketing strategies and enhancing customer experiences. The system's capability to analyze facial expressions also assists in medical diagnosis and treatment, facilitating early intervention and personalized care.

The project has taken significant steps to advance face detection technology, but it is essential to acknowledge its limitations. Variability in environmental conditions, computational resource constraints, and ethical considerations pose challenges that must be addressed to ensure the system's effectiveness and ethical deployment. This project represents a significant step forward in face detection and advances data science principles to develop a robust and efficient system. By addressing challenges and exploring practical applications, we have put the basis for future advancements in this area, with the potential to make a meaningful impact crosswise various domains.

Several capable paths have developed to further advance the face detection field and extend our system's capabilities. These future directions include both technical innovations and broader applications in diverse domains. One key direction for future research involves enhancing the robustness and adaptability of face detection systems to address the challenges posed by variable environmental situations. By developing algorithms proficient in automatically adjusting to variations in lighting, facial expressions, and environmental settings, we can improve the accuracy and reliability of face detection in real-world scenarios. Exploring novel methods such as deep learning techniques and ensemble systems will offer new perceptions of optimizing system performance and generalizing different datasets and environments.

Integrating multimodal data sources, such as audio and video streams, can enhance the contextual understanding of facial connections and expressions. Merging facial recognition with other sensory systems allows us to develop more comprehensive and advanced models which capable of capturing intangible indications and distinctions in human behavior. This approach opens up prospects for applications in fields such as human-computer interaction, sentimental computing, and social robotics, where understanding facial expressions plays a crucial role in simplifying natural and sympathetic communications. Another important direction for future research involves addressing ethical and social considerations surrounding the deployment of face detection systems. As facial recognition technology becomes increasingly dominant in various characteristics of humanity, it is essential to arrange confidentiality, agreement, and answerable use to moderate potential risks and biases. Future research efforts should focus on developing transparent and accountable frameworks for the ethical design, deployment, and instruction of face detection systems, ensuring that they adhere to ethical principles and legal regulations while minimizing potential harms and biases.

Exploring face detection applications beyond traditional domains such as security and surveillance offers exciting opportunities for innovation and societal impact. For example, in healthcare, integrating facial recognition technology with telemedicine platforms can facilitate remote patient monitoring and enhance the delivery of personalized care. Similarly, in education, facial recognition for adaptive learning systems can provide personalized feedback and support to students based on their learning styles and needs. Future research directions in face detection incorporate various technical innovations and applications across multiple domains. By enhancing system robustness, addressing ethical considerations, and exploring novel applications, we can continue to push the boundaries of what is possible with face detection technology and unlock its full potential to benefit society.

# References

Akhtar, I., Mudawi, N. Al, Alabdullah, B.I., Alonazi, M., Park, J., 2023. Human-Based Interaction Analysis via Automated Key Point Detection and Neural Network Model. IEEE Access 11. https://doi.org/10.1109/ACCESS.2023.3314341

Akhter, I., Jalal, A., 2023. Abnormal Action Recognition in Crowd Scenes via Deep Data Mining and Random Forest, in: 2023 4th International Conference on Advancements in Computational Sciences, ICACS 2023 - Proceedings. https://doi.org/10.1109/ICACS55311.2023.10089674

Akhter, I., Javeed, M., 2022. Pedestrian Behavior Recognition via a Smart Graph-based Optimization, in: 2022 19th International Bhurban Conference on Applied Sciences and Technology (IBCAST). pp. 629–634. https://doi.org/10.1109/IBCAST54850.2022.9990434

Alam, A., Abdullah, S.A., Akhter, I., Alsuhibany, S.A., Ghadi, Y.Y., al Shloul, T., Jalal, A., 2022. Object detection learning for intelligent self automated vehicles. INTELLIGENT AUTOMATION AND SOFT COMPUTING 34, 941–955.

Aljuaid, H., Akhter, I., Alsufyani, N., Shorfuzzaman, M., Alarfaj, M., Alnowaiser, K., Jalal, A., Park, J., 2023. Postures anomaly tracking and prediction learning model over crowd data analytics. PeerJ Comput Sci 9. https://doi.org/10.7717/peerj-cs.1355

Beyan, C., Vinciarelli, A., Del Bue, A., 2022. Face-to-Face Co-Located Human-Human Social Interaction Analysis using Nonverbal Cues: A Survey. arXiv preprint arXiv:2207.10574.

Bonettini, N., Cannas, E.D., Mandelli, S., Bondi, L., Bestagini, P., Tubaro, S., 2021. Video face manipulation detection through ensemble of cnns, in: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 5012–5019.

Canny, J., 1986. A Computational Approach to Edge Detection. IEEE Trans Pattern Anal Mach Intell PAMI-8. https://doi.org/10.1109/TPAMI.1986.4767851

Cheddad, A., Condell, J., Curran, K., Mc Kevitt, P., 2009. A skin tone detection algorithm for an adaptive approach to steganography. Signal Processing 89. https://doi.org/10.1016/j.sigpro.2009.04.022

Chen, D., Ren, S., Wei, Y., Cao, X., Sun, J., 2014. Joint cascade face detection and alignment, in: European Conference on Computer Vision. pp. 109–122.

Chen, H.-W., McGurr, M., 2016. Moving human full body and body parts detection, tracking, and applications on human activity estimation, walking pattern and face recognition, in: Automatic Target Recognition XXVI. https://doi.org/10.1117/12.2224319

Chua, T.-S., Zhao, Y., Kankanhalli, M.S., 2002. Detection of human faces in a compressed domain for video stratification. Vis Comput 18, 121–133.

Dada, E.G., Joseph, S.B., Oyewola, D.O., Fadele, A.A., Chiroma, H., Abdulhamid, S.M., 2022. Application of Grey Wolf Optimization Algorithm: Recent Trends, Issues, and Possible Horizons. Gazi University Journal of Science 35. https://doi.org/10.35378/gujs.820885

De-La-Torre, M., Granger, E., Radtke, P.V.W., Sabourin, R., Gorodnichy, D.O., 2015. Partially-supervised learning from facial trajectories for face recognition in video surveillance. Information Fusion 24. https://doi.org/10.1016/j.inffus.2014.05.006

Dong, Z., Wei, J., Chen, X., Zheng, P., 2020. Face Detection in Security Monitoring Based on Artificial Intelligence Video Retrieval Technology. IEEE Access 8. https://doi.org/10.1109/ACCESS.2020.2982779

Ghadi, Y., Akhter, I., Alarfaj, M., Jalal, A., Kim, K., 2021. Syntactic model-based human body 3D reconstruction and event classification via association based features mining and deep learning. PeerJ Comput Sci 7, e764.

Ghadi, Y.Y., Akhter, I., Aljuaid, H., Gochoo, M., Alsuhibany, S.A., Jalal, A., Park, J., 2022a. Extrinsic Behavior Prediction of Pedestrians via Maximum Entropy Markov Model and Graph-Based Features Mining. Applied Sciences 12. https://doi.org/10.3390/app12125985

Ghadi, Y.Y., Akhter, I., Alsuhibany, S.A., al Shloul, T., Jalal, A., Kim, K., 2022b. Multiple Events Detection Using Context-Intelligence Features. Intelligent Automation \& Soft Computing 34, 1455–1471.

Gochoo, M., Akhter, I., Jalal, A., Kim, K., 2021. Stochastic Remote Sensing Event Classification over Adaptive Posture Estimation via Multifused Data and Deep Belief Network. Remote Sens (Basel) 13. https://doi.org/10.3390/rs13050912

Gupta, A., Tiwari, R., 2014. Face detection using modified Viola jones algorithm. International Journal of Recent Research in Mathematics Computer Science and Information Technology 1, 59–66.

Jalal, A., Akhtar, I., Kim, K., 2020. Human Posture Estimation and Sustainable Events Classification via Pseudo-2D Stick Model and K-ary Tree Hashing. Sustainability 12, 9814.

Jiang, H., Learned-Miller, E., 2017. Face detection with the faster R-CNN, in: 2017 12th IEEE International Conference on Automatic Face \& Gesture Recognition (FG 2017). pp. 650–657.

Korshunov, P., Marcel, S., 2018. Deepfakes: a new threat to face recognition? assessment and detection. arXiv preprint arXiv:1812.08685.

Kumar, S., Singh, S., Kumar, J., 2018. Live Detection of Face Using Machine Learning with Multi-feature Method. Wirel Pers Commun 103. https://doi.org/10.1007/s11277-018-5913-0

Li, L., Bao, J., Zhang, T., Yang, H., Chen, D., Wen, F., Guo, B., 2020. Face x-ray for more general face forgery detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5001–5010.

Lu, C., Shi, J., Jia, J., 2013. Abnormal event detection at 150 fps in matlab, in: Proceedings of the IEEE International Conference on Computer Vision. pp. 2720–2727.

Mallick, A.K., Verma, A., Sahay, U., Shubham, K., Jayanna, H.S., 2020. Artificial intelligence based video monitoring system for security applications, in: 11th International Conference on Advances in Computing, Control, and Telecommunication Technologies, ACT 2020.

Mehmood, F., Ullah, I., Ahmad, S., Kim, D.H., 2019. Object detection mechanism based on deep learning algorithm using embedded IoT devices for smart home appliances control in CoT. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-019-01272-8

Messikh, N., Bousba, S., Bougdah, N., 2017. The use of a multilayer perceptron (MLP) for modelling the phenol removal by emulsion liquid membrane. J Environ Chem Eng 5. https://doi.org/10.1016/j.jece.2017.06.053

Montserrat, D.M., Hao, H., Yarlagadda, S.K., Baireddy, S., Shao, R., Horváth, J., Bartusiak, E., Yang, J., Guera, D., Zhu, F., others, 2020. Deepfakes detection with automatic face weighting, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 668–669.

Ojala, R., Vepsäläinen, J., Tammi, K., 2022. Motion detection and classification: ultra-fast road user detection. J Big Data 9. https://doi.org/10.1186/s40537-022-00581-8

Pervaiz, M., Akhter, I., Chelloug, S.A., 2022. An Optimized System for Human Behaviour Analysis in E-Learning, in: 2022 International Conference on Electrical Engineering and Sustainable Technologies, ICEEST 2022 - Proceedings. https://doi.org/10.1109/ICEEST56292.2022.10077871

Sabir, E., Cheng, J., Jaiswal, A., AbdAlmageed, W., Masi, I., Natarajan, P., 2019. Recurrent-convolution approach to deepfake detection-state-of-art results on faceforensics++. arXiv preprint arXiv:1905.00582.

Ullah, R., Hayat, H., Siddiqui, A.A., Siddiqui, U.A., Khan, J., Ullah, F., Hassan, S., Hasan, L., Albattah, W., Islam, M., Karami, G.M., 2022. A Real-Time Framework for Human Face Detection and Recognition in CCTV Images. Math Probl Eng 2022. https://doi.org/10.1155/2022/3276704

Viola, P., Jones, M.J., 2004. Robust Real-Time Face Detection. Int J Comput Vis. https://doi.org/10.1023/B:VISI.0000013087.49260.fb

Wang, F., 2022. Application of artificial intelligence-based video image processing technology in security industry. https://doi.org/10.1117/12.2642729

Wibowo, H.T., Prasetyo Wibowo, E., Harahap, R.K., 2021. Implementation of Background Subtraction for Counting Vehicle Using Mixture of Gaussians with ROI Optimization, in: 2021 6th International Conference on Informatics and Computing, ICIC 2021. https://doi.org/10.1109/ICIC54025.2021.9632950

Yao, Y., Yang, G., Sun, X., Li, L., 2016. Detecting video frame-rate up-conversion based on periodic properties of edge-intensity. Journal of Information Security and Applications 26. https://doi.org/10.1016/j.jisa.2015.12.001

Zhou, P., Han, X., Morariu, V.I., Davis, L.S., 2017. Two-stream neural networks for tampered face detection, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1831–1839.

# Chapter 8: Appendices

1. Data Preprocessing:

The project started with data preprocessing, a fundamental step in preparing video-based data for analysis. This involved converting videos into frame sequences, resizing the images to a uniform size, and reducing noise using techniques like median filtering. Standardizing the data and reducing noise laid the groundwork for more accurate feature extraction in subsequent stages.

2. Feature Extraction:

After preprocessing, the focus shifted to extracting the region of interest (ROI) from the images. Techniques such as skin tone, edge, and motion detection were employed to identify and isolate the key regions containing facial features. This step aimed to enhance the system's ability to focus on relevant feature extraction and classification areas.

3. Facial Feature Extraction:

Feature extraction was a critical step in capturing distinctive facial characteristics from the ROI.

4. Classification Using MLP:

The final step involved classifying facial patterns based on the extracted features. The Multilayer Perceptron (MLP) algorithm was employed for categorical classification, leveraging machine learning principles to discern subtle nuances in facial characteristics. By training the MLP with labelled data and fine-tuning its parameters, the system could accurately classify facial patterns with high precision.

5. Data Optimization Using GWO:

Data optimization involved fine-tuning the system's parameters to improve its performance and adaptability.

6. Comparison of MLP and RF:

In the comparison of both Multilayer Perceptron (MLP) and Random Forest (RF) classifiers, we evaluated their performance based on various metrics, including accuracy, precision, recall, and F1-score. MLP demonstrated slightly higher accuracy (98.0%) than RF (97.5%), making it suitable for tasks with high-dimensional data and intricate patterns. However, RF offers robustness and interpretability, making them complementary approaches in machine learning classification tasks.

7. Limitations:

- Variability in Environmental Conditions.
- Computational Resources.
- Ethical Considerations.
- Generalization.

8. Applications:

- Security and Surveillance.
- Automated Attendance Systems.
- Customer Analytics.
- Healthcare.

9. Acknowledgment:

We want to express our sincere gratitude to all who contributed to this project, including our supervisor, colleagues, and participants. Their support and assistance were invaluable in the successful completion of this research.

10. Meeting history and schedule with the Supervisor

| Sr | Meeting date | Time | Location | Detail |
|---|---|---|---|---|
| 1 | 14/12/2023 | 15:00 | Room LE 123() Headingly | Discussion about the topic selection, Approval, and overall strategy during the supervision |
| 2 | 30/12/2023 | 16:00-17:00 | Online | Discussion about the research paper. The 4 paper I have used throughout my project is my base paper. Discuss Zotero software use for referencing.<br>Thematic review.<br>Read much<br>Write in your word<br>Literature search |
| 3 | 02/02/2024 | email | Through email | Discussion ABOUT ethical approval and process |

| | | | | |
|---|---|---|---|---|
| 4 | 08/02/2024 | 09:00-10:00 | Room LE 123() Headingly | Feedback about the proposal. Some AI text was detected; discuss the data I used in the project. |
| 5 | 01/03/2024 | 15:30-16:30 | Online | Change in the title was very lengthy |
| 6 | 03/03/2024 | 09:00-10:00 | Online | Ethical approval/ some hurdle/ discussion about the online form of |
| 7 | 01/04/2024 | Email | Email | Discussion about the draft chapter Methodology chapter suggest. |
| 08 | 26/04/2024 | 17:00-18:00 | Online | Discussion about the draft chapter feedback, poster discussion, discussion showing some result/progress/coding Overall, it shows my progress. |
| 09 | 04/05/2024 | Email | Email | Contact through email, share report for review, |
| 10 | 05/05/2024 | | | Share the poster for the review. Address the suggestion and make it again. |

Screenshot of the code:

```
# Mount Google Drive
from google.colab import drive
drive.mount('/content/drive', force_remount=True)
# Import necessary libraries
import cv2
# Path to the video file on your Google Drive
video_path = '/content/drive/MyDrive/Chap_3/01.avi'
# Open the video file
video_capture = cv2.VideoCapture(video_path)
# Check if the video opened successfully
if not video_capture.isOpened():
    print("Error: Unable to open video.")
else:
    # List to store frames
    frames = []
    # Read and store each frame of the video
    while True:
        # Read a frame from the video
        ret, frame = video_capture.read()
        # If no frame is read, break the loop
        if not ret:
            break
```

*Figure 8-1 Sample of code-1*

```
# Detect heads in a frame
import cv2
import numpy as np
import matplotlib.pyplot as plt
from google.colab.patches import cv2_imshow

# Function to detect heads in a frame
def detect_heads(frame):
    # Convert the frame to grayscale
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Apply background subtraction to extract moving objects (potential heads)
    fg_mask = fgbg.apply(frame)

    # Apply morphological operations to enhance the mask
    fg_mask = cv2.morphologyEx(fg_mask, cv2.MORPH_OPEN, kernel)
    fg_mask = cv2.morphologyEx(fg_mask, cv2.MORPH_CLOSE, kernel)

    # Find contours in the foreground mask
```

*Figure 8-2 Sample of code-2*

```
    plt.imshow(edge_detected_frames[idx], cmap='gray')
    plt.title(f'Edge-Detected Frame {idx}')
    plt.axis('off')

# Show the average result
plt.subplot(num_frames_to_show, 2, num_frames_to_show*2-1)
plt.imshow(avg_edge_detected_frame, cmap='gray')
plt.title('Average Edge-Detected Frame')
plt.axis('off')

plt.show()

# Show the frame count
print("Total Filtered Frames:", len(median_filtered_frames))
```

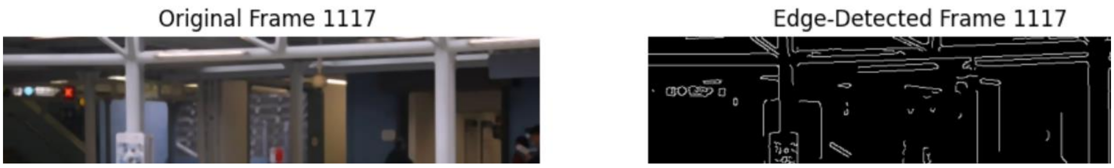


*Figure 8-3 Sample of code-3*

```python
# Motion detection
def apply_motion_detection_all_frames(frames):
    motion_detected_frames = []
    prev_frame = cv2.cvtColor(frames[0], cv2.COLOR_BGR2GRAY)

    for frame in frames[1:]:
        # Convert frames to grayscale
        current_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        # Compute absolute difference between frames
        diff = cv2.absdiff(prev_frame, current_frame)

        # Apply threshold to detect motion
        _, motion_detection = cv2.threshold(diff, 30, 255, cv2.THRESH_BINARY)

        # Store the motion-detected frame
        motion_detected_frames.append(motion_detection)

        # Update previous frame
        prev_frame = current_frame
```

*Figure 8-4 Sample of code-4*

```python
[ ] # Initialize an empty list to store change detection values
    change_detection_values = []

    # Loop through all frames for change detection
    for i in range(len(median_filtered_frames) - 1):
        # Perform change detection between consecutive frames
        change_detected_frame = change_detection(median_filtered_frames[i], median_filtered_frames[i + 1])

        # Calculate the sum of change detection values for each frame
        change_detection_sum = np.sum(change_detected_frame)

        # Append the sum to the list
        change_detection_values.append(change_detection_sum)

    # Convert the list to a NumPy array
    change_detection_values = np.array(change_detection_values)
```

*Figure 8-5 Sample of code-5*

```python
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Function to calculate motion flow vectors and distance features
def calculate_motion_flow_and_distance(frames):
    # Initialize parameters for Lucas-Kanade optical flow
    lk_params = dict(winSize=(15, 15),
                     maxLevel=2,
                     criteria=(cv2.TERM_CRITERIA_EPS | cv2.TERM_CRITERIA_COUNT, 10, 0.03))

    # Initialize lists to store motion flow vectors and distance features
    motion_flow_vectors = []
    distance_features = []

    # Convert the first frame to grayscale
    prev_frame = cv2.cvtColor(frames[0], cv2.COLOR_BGR2GRAY)

    # Iterate through each frame starting from the second frame
    for i in range(1, len(frames)):
        # Convert the current frame to grayscale
        current_frame = cv2.cvtColor(frames[i], cv2.COLOR_BGR2GRAY)
```

*Figure 8-6 Sample of code-6*

```python
import numpy as np
import matplotlib.pyplot as plt
# Plot motion flow vectors and distance features
plt.figure(figsize=(12, 6))
# Plot motion flow vectors
plt.subplot(1, 2, 1)
average_flow = np.mean(motion_flow_vectors, axis=0)
plt.imshow(average_flow, cmap='hsv', interpolation='nearest')
plt.title('Average Motion Flow Vectors')
plt.colorbar()
plt.axis('off')

plt.show()
```

*Figure 8-7 Sample of code-7*

```python
from google.colab import drive
import pandas as pd

# Mount Google Drive
drive.mount('/content/drive')

# Path to the CSV file on Google Drive
file_path = '/content/drive/MyDrive/Chap_3/dataset_normal_and_abnormal.csv'

# Read the CSV file into a DataFrame
df = pd.read_csv(file_path)

# Display the header of the DataFrame
print(df.head())
```

```
Mounted at /content/drive
     motion    change    distance     Label
0  4.373570  0.483224   88.709425    normal
1  7.280921  0.297178   43.169311    normal
2  5.768075  0.552126   33.570048    normal
3  6.574027  1.707691  105.837245  abnormal
```

*Figure 8-8 Sample of code-8*

```python
# Data normalization
df_normalized = df.copy()
features = ['motion', 'change', 'distance']
for feature in features:
    df_normalized[feature] = (df_normalized[feature] - df_normalized[feature].min()) / (df_normalized[feature].max() - df_normalized[feature].min())

# Plot data normalization
plt.figure(figsize=(12, 6))
for feature in features:
    sns.kdeplot(df_normalized[feature], label=feature)
plt.title('Data Normalization Plot')
plt.xlabel('Normalized Value')
plt.ylabel('Density')
plt.legend()
plt.show()
```

*Figure 8-9 Sample of code-9*

```
# Train the model and store accuracy values
train_accuracies = []
test_accuracies = []
for i in range(1, 101):
    mlp.partial_fit(X_train, y_train, classes=np.unique(y_train))
    train_accuracies.append(accuracy_score(y_train, mlp.predict(X_train)
    test_accuracies.append(accuracy_score(y_test, mlp.predict(X_test)))

# Plot accuracy curve
plt.figure(figsize=(10, 6))
plt.plot(range(1, 101), train_accuracies, label='Train Accuracy')
plt.plot(range(1, 101), test_accuracies, label='Test Accuracy')
plt.xlabel('Number of Iterations')
plt.ylabel('Accuracy')
plt.title('Accuracy Curve')
plt.legend()
plt.show()
```

*Figure 8-10 Sample of code-10*