I.J. Wireless and Microwave Technologies, **2025, 2, 26-40**
Published Online on April 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijwmt.2025.02.03

Android Mobile Security and File Protection Using Face Recognition

Marah Radi Hawa
Department of Natural, Engineering and Technology Sciences, Arab American University (AAUP), Ramallah, Palestine
E-mail: m.hawa1@student.aaup.edu
ORCID ID: https://orcid.org/0009-0006-2205-7886

Amani Yousef Owda*
Department of Natural, Engineering and Technology Sciences, Arab American University (AAUP), Ramallah, Palestine
E-mail: amani.owda@aaup.edu
ORCID ID: https://orcid.org/0000-0002-6104-9508
*Corresponding author

Majdi Owda
Faculty of Artificial Intelligence and Data Science, UNESCO Chair on Data Science for Sustainable Development, Arab American University (AAUP), Ramallah, Palestine
E-mail: majdi.owda@aaup.edu
ORCID ID: https://orcid.org/0000-0002-7393-2381

Received: 27 August, 2024; Revised: 13 November, 2024; Accepted: 08 January, 2025; Published: 08 April, 2025

Abstract: The use of Android devices has increased rapidly in recent years, increasing the chance of hacking and crime. Hackers target smartphones for various purposes, including getting sensitive information, financial fraud, identity theft, and other crimes. As a result, Android users must be aware of these possible dangers and take necessary measures to secure their smartphones. Because smartphones are the primary repository of personal sensitive information, smartphone designers must include security measures and encourage users to install freely available security software. Most studies have evaluated facial recognition as the most secure feature. This paper shows the uses of a facial recognition application to protect user files that contain sensitive information. The application uses machine-learning algorithms, specifically a Convolutional Neural Network (CNN) for face recognition that detects the user's face, tries to access the file, compares it with the basic image in the local file, and gives the result of whether to open the file or reject depending on the compared image. The application addresses critical concerns and improves file privacy features on Android devices, ensuring user file safety, and achieving success with 99% accuracy. It can also distinguish the faces of women wearing a shawl and people wearing glasses.

Index Terms: Android Smartphones, Security Software, Facial Recognition, Android File Security

1. Introduction

In these days, smartphones have become more than just communication devices. They have become an integral part of our lives, and smartphones have developed into a significant repository of personal information, including bank accounts and contact details. Therefore, smartphone manufacturers are putting effort into enhancing security features, and the latest mobile phones have advanced features to safeguard our data. These devices offer an array of capabilities that were unimaginable a decade ago, making them powerful computing platforms that can handle complex tasks, from browsing the internet to running advanced applications. This makes files vulnerable to hacking by unauthorized people [1,2].

Smartphones on the Android operating system come equipped with several basic applications such as contacts, cameras, radios, memos, alarms, and many other features. This makes them extremely convenient and popular, encouraging people to use them more often in their free time or whenever they get the chance.

They are also widely used for networking through chat, email, and social media. Additionally, smartphones are used for processing tasks, real-time communication, and multimedia services, including multimedia data sharing and video conferencing. Due to these factors, the use of smartphones has exploded over the years. However, because of

their advanced features, they are often carried everywhere, making them susceptible to theft or loss, as well as various types of attacks. Consequently, most mobile devices come equipped with built-in locking mechanisms, such as Personal Identification Number (PIN) passwords, patterns, fingerprint recognition, and sliding lock (swiping left or right).

Although these are the most commonly used locking mechanisms, they are unfortunately vulnerable to shoulder surfing and smudge attacks. Therefore, new and effective locking approaches must be developed [1]. To address these issues, it is important to:

- To investigate whether personal data and files on the device are secure from unauthorized access.
- To protect against data breaches and theft.
- To give consumers a smooth and simple experience by utilizing a safe and quick authentication technique.

The main contribution of this paper is the development of a novel machine-learning algorithm for file protection on Android smartphones. The application aims to reduce attacks on Android files and protect user data. In addition, the algorithm can distinguish between people wearing eyeglasses and those using face recognition without eyeglasses.

This section provides an overview of Android file attacks and the remainder of this paper is organized as follows: Section 2 describes the literature review and previous work. Section 3 describes the methodology that was employed. Section 4 illustrates the results and compares the model proposed in this paper and those from other studies in the literature. Section 5 summarizes the overall conclusions of the paper.

2. Background

In 2023, approximately 3 billion smartphone users are predicted to be Android users, with at least 80% of them [3]. Asia, as one of the leading countries in smartphone development, has the most Android users compared to the Americas, Europe, and Australia [4], including Japan, India, Singapore, and China. India is the fastest-growing smartphone country, with at least 100% of its population using smartphones [3], With these statistics, an increasing number of users are subject to personal information and sensitive data experiences. 78% of smartphone users had their identities, such as their names, private information, pictures, and classified movies hacked [5]. There are many variables affecting mobile phone security. This literature review is divided into the following sections: Section 2.1 Security Authentication, section 2.2 Face Recognition Development, and Section 2.3 Android Application.

2.1 Security Authentication

Mobile phones frequently do not have passwords to identify users and manage access to data maintained on the devices. This increases the risk of unauthorized access to sensitive information on stolen or lost phones [6]. In other cases, authentication occurs due to a simple pattern or Personal Identification Number (PIN) [7] that can be predicted, forgotten, written down stolen, or eavesdropped [8]. To avoid tracking, most users turn off the phone's location tracking [9], restricting its capabilities in added security if it is stolen or forgotten [10]. The existing security app lock features a simple set of authentications, which results in the same degree of security as Android itself offers [11]. Other apps have a complex set of authentications that take a long time for customers to open their phones and applications [4]. Furthermore, the majority of program locks on the market may be simply deactivated or removed [12]. Most app locks' weak security authentication can still risk the protection intended by the user upon download [13]. The human face gives a lot of information about a person's identity. Facial recognition is being developed to capitalize on its uniqueness [14]. Facial recognition provides the most secure solution because each individual is distinct from others. One of the most active research areas in secure information systems is biometric identification on smartphones. A great deal of research has been published on the creation of facial recognition apps for Android smartphones. Connection attackers. An effective server or virtual server assault will harm all service elements aggregated under the affinity protocols. In terms of detection, attacks on one service instance will likely affect others in the same affinity group. Affinity policies might thus be utilized as an early warning system to prevent attack transmission throughout numerous services. Unfortunately, there is no standard means for cloud service suppliers or particular renters to quickly share this information with other companies.

2.2 Face Recognition Development

Much research has been published on face recognition development for Android smartphones.

A study by Guillaume Dave et al. [15] focused on the performance of face recognition algorithms on smartphones. The researchers experiment with the algorithms on smartphones with 600 MHz processors and 256 Mb RAM. By using 134 face images of 10 different people to make their tests. The results showed that using the Fisher face algorithm, the recognition rate was 94%, achieved within 1.6 seconds.

Vazquez-Femandez et al. [16] demonstrated a smart picture-sharing app for Android devices that uses facial recognition. High Tech Computer Corporation (HTC) Desire with 1 GHz engine and 576 MB RAM and Samsung Galaxy Tab with 1 GHz processing and 512 MB RAM were used in the tests. They checked the application for 50 contacts with four pictures for each interaction and obtained 0.35-sex on HTC Desire and 0.47 sec on Samsung Galaxy Tab to recognize the facial features.

Dospinescu and Popa [17] developed an Android application that uses face recognition technology to allow access to specific regions or rooms only to legitimate people. This application can be useful in various settings such as hospitals or educational institutions where only employees are allowed to enter certain rooms. The authors utilized the Face identity algorithm by importing the OpenCV library into the Android project for the facial recognition process. They evaluated the application's face recognition and detection functions by evaluating it on different types of photos which include 1 to 100 faces. The accuracy of the results is affected by the lighting conditions and the camera's face location. Face detection is a highly efficient technique that can detect multiple faces in a group picture. In the event of a group of pictures, still, the face detection technique is more reliable than the facial recognition process.

Chaudhry and Chandra [18]. Explored mobile computing to develop a face recognition and identification system for visually asked individuals in their work. The technology makes use of the camera and earpiece of a smartphone's mobile device to create a small and lightweight device, which is aided by a server-based assistance system. The facial recognition technology is user-friendly for visually impaired people, with a simple interface. The authors executed the built-in algorithms of the OpenCV library into the Android project for face recognition. For quick extraction of features, the system employs the Local Binary Pattern Histograms (LBPH) algorithm. They put the smartphone app through its paces on a bespoke video database using eight films shot in the nighttime with varying facial jobs. The face detection experiment was repeated 80 times, with results indicating an accuracy of up to 93% in well-lit situations. When the individual stared squarely at the camera with a neutral expression, it achieved greater detection accuracy. Overall, this work gives a possible way to allow visually requested people to swiftly identify and locate faces. When faces can be identified at narrow distances and different facial expressions are utilized, accuracy decreases. Over 50 test runs for face recognition. Persons gazing virtually directly at the camera with a neutral face look had fairly good recognition accuracy. Recognizing faces at various positions with diverse facial expressions produced lower results. The usage of solely frontal visage images for joining a person into the face registry is a major factor in low accuracy in identification.

The OpenCV library is used to recognize and detect faces within the mobile application [19]. The Local Binary Pattern face features classifier was used for face detection, whereas the Local Binary Pattern Histograms model was used for face recognition. Although the recognition algorithms perform well, they are influenced by a variety of considerations like as the lighting, the person's perspective on facial expressions, face covering, camera features, and the operation of the mobile device itself.

When algorithms are used for facial recognition, their performance is affected by a variety of variables such as lighting, the person's position, expressions, face coverage, camera features, and the mobile device's performance. Marian Harbach et al. [20] evaluate the efficiency of smartphone keys in their research. They attempt to develop solutions that prevent unwanted parties from obtaining access to devices (security), while simultaneously reducing the load on legitimate users (usability). Over a single month, 134 people who had instrumented smartphones logged incidents. The findings demonstrated that current lock screen technologies provide customers with significant compromises between usability (discovering speed vs. unlocking frequency) and security. Pattern users, who opened more frequently and were more prone to errors, took longer to enter their codes but made fewer errors. On average, PIN and pattern users require an identical amount of time to get into their devices. However, there have been limitations to the findings presented in this study. To begin, all participants were University of Buffalo pupils or staff, so they may not be representative of all people who use smartphones. Second, every participant used the same device, an LG Nexus 5. Finally, the authors' data collection period exceeded thirty days.

Many people recently saved crucial files on their smartphones. According to the findings of statistics in a survey [21], 24.12%of Android users use Smart Lock; 32.35% use CM Lock; one-hundred twenty- 36.47%use AppLocker; 4.12% use Finger security; 0.29% use Privacy Knight; and 2.65% do not use any other application. Based on these outcomes, the majority of users used AppLocker, indicating that researchers should focus more on securing programs on Android smartphones as safety is absent for apps in other security apps, most of the applications available in the market can be easily removed or uninstalled and have some limitation.

Tan and Triggs [22] created a system for recognizing faces in uncontrolled lighting settings. To reduce light effects while maintaining essential image details, they used powerful lighting normalization, local texture-based representation, altered distance transformation, kernel-based feature extraction, and a variety of features. Local Ternary Patterns (LTP) were introduced, which are less sensitive to noise and more effective than Local Binary Patterns (LBP). They also modified their methods by incorporating Gabor wavelets with LBP and obtaining attributes using Kernel Principal Component Analysis (PCA). On the FRGC-204 dataset, they attained an accuracy rate of 88.1% with a false acceptance rate of 0.1%.

Another option for unlocking a smartphone is to develop a mechanism that uses variations in facial expressions to address the problem of computers being unable to discriminate between genuine faces and photos [23]. The locked smartphone will be opened once the face has been correctly recognized and the way it looks has been adjusted [23].

Our Android application for securing files will differentiate itself by addressing critical concerns regarding file privacy and safety on Android devices. With an impressive success rate of 99% accuracy, our application not only secures files but also utilizes advanced facial recognition technology to ensure that only authorized users can access sensitive information. Notably, it can accurately distinguish the faces of individuals wearing shawls or glasses, providing a robust layer of security in diverse situations. By integrating these advanced features, our application aims to significantly enhance file privacy, giving users peace of mind knowing their data is well protected against unauthorized access.

2.3 Android Application

A group of developers have developed a creative Android application that employs powerful algorithms for facial recognition to secure the phone being used by the user from possible hackers. This advanced technology scans the user's face with the smartphone's front-facing camera and only allows access to the device if the facial features match those registered by the owner.

The researchers in[21] provided a survey to 340 respondents, asking about how often users change the safety applications' passwords. 1.76%change their password on a regular or daily basis; 24.71%) change their password every week; one hundred forty-five (145, 42.65% change their password every month; 10.59% change their password every year; and 11.47% rarely change their password. The majority of respondents changed their password (PIN, Pattern, and Password) monthly, indicating that they are not happy with their security and need to update regularly. As Android smartphone's PIN, Pattern, and Password can be easily determined, this highlights the need to develop an application that can protect user smartphones and files.

F-Loker application [21]. This is capable of providing security to applications selected by the users. It enables you to "Start Service" and "Block" particular apps, including built-in and downloaded applications. The system is ready to use once the safety options are chosen. Users just need to choose all of the installed programs mentioned in the system to secure and then click "save." Face recognition security is applied to all specified applications before they can be opened. If the face recognition fails on three occasions in succession to match the authorized user's face, access to the application fails and the application must close.

App Locker in[1] is an Android application that secures your device's screen and application access. It uses the LBPH algorithm from the OpenCV library to deliver rapid and efficient face recognition. The face recognition method works perfectly even in low-light situations. App Locker protects your entire phone with a single app, removing the need to install third-party apps or utilize the settings app. Furthermore, the app offers an online interface from which you or a system administrator may view all installed applications, determine which are locked and which are not, as well as enable or disable others. This capacity is particularly useful in company settings where employees are bound to certain limitations.

File Management application in[24] includes all of the standard file management operations, such as recording, slicing, pasting, deleting, renaming, compressing, decompressing, transferring, downloading, and bookmarking. It allows you to browse files not only on your devices but also across your network, including FTP, SMB, and cloud storage. It supports an extensive selection of media files and major file formats, including[24].

AppLocker with Face is illustrated in Fig.1.[25] is an app that uses facial recognition technology to lock any app on your phone shown in Fig.2.This clever biometric function scans your face, ensuring that only you have access to the apps you have limited access to. You may apply the lock to any app on your phone, adding another layer of security to your personal information.

Fig. 1. AppLocker with Face [25].

Super Applock with Face ID in illustrated Fig.2.[26] is an app that allows you to lock and unlock apps using your face. That is simple and safeguards your mobile phone's apps. If you see any untrue advertisements in Face ID, Face ID is fully dedicated to combating them. If you are already familiar with applications with face lock or face tracking, then you will be interested in our own.

Fig. 2. Super AppLocker with Face [26].

Smart Lock app [27] protects your mobile phone's privacy from hackers. You can find the most essential app here that conceals apps/pictures/media files by managing passwords. It offers many advantages, such as app locking, becoming awake, and auto screen rotation.

2.4 Recent Advanced Face Recognition Technology

Facial recognition technology has seen a significant development in recent years, driven by the growing need for reliable biometric authentication solutions, especially in security applications on mobile devices [28]. Over the past decade [28,29], models have shifted from relying on manually defined features to deep learning, which has helped improve model performance under diverse environmental conditions such as lighting and facial angles. Advances in convolutional neural networks (CNNs) have contributed to a high resolution approaching the human level of identity verification [28]. These models are designed to handle various challenges, making them suitable for security applications such as file protection on Android systems. According to recent research, these include significant advantages in technologies such as DeepFace and ResNet [30], which research suggests better concealment of agents, boosting performance in hardware-specific applications.

2.5 Deep Learning Approaches in Face Recognition

Deep learning has greatly contributed to improving facial recognition technology, as convolutional neural networks (CNNs) have improved accuracy in recognizing complex features in images [31]. For example, networks such as ResNet and ArcFace have dramatically transformed the performance of models by extracting fine details from facial images [32].ResNet introduced the residual connection method, allowing for increasing network depth and improving its ability to extract rich representations that could resist challenges such as changes in lighting and facial angles [32]. In addition, ArcFace has proven effective in improving identity recognition by adopting additional angle loss that enhances the distinction of nuances between people [30]. These technologies are essential for facial recognition applications in mobile devices, as they provide robust and reliable protection without compromising performance, making them ideal for security applications such as file protection in Android [30].

Our application in this paper is developed to provide an extra layer of security to the user's device, and it can be useful if the device is lost or stolen. It can also prevent unwanted access by thieves seeking to hack into the device. This paper will answer the following research questions:

1- Can machine learning algorithms be effectively utilized to develop a model for an Android application locker?
2- Is the application providing greater accuracy and giving distinct outcomes?
3- Is the face recognition application enhanced security and user-friendly features?

3. Methodology

This study aims to develop an android application utilizing facial recognition technology to enhance smartphone security, by using machine learning algorithms specifically deep convolutional neural networks for different kinds of pictures. Then, in evaluating the suggested application's performance, we compare the accuracy of our proposed application to that of current applications by gathering input from Android users on applications. This study utilizes a mix of research methods. Qualitative research methods for facial recognition, and quantitative research to assess the performance metrics of the machine-learning algorithms presented in this paper.

We can divide the face recognition operation into three parts: face detection, comparing faces with the faces saved in the system, and face recognition with the goal. These three parts are explained in detail in the following phases, as shown in the accompanying image.

Fig.3. depicts the main phases of the implemented Android application for securing files. The next subsections within this section will provide detailed insights into these six phases and their inside components utilized in constructing the proposed application for protecting files.

Fig. 3. Main phases of Android application to protect files.

3.1 Image Collection

The initial step involves the use of a mobile application developed using the Flutter framework. The app enables users to capture their facial images using the camera on their phones, leveraging the "camera" package to ensure the efficient real-time capture of images. Subsequently, the captured images are transmitted to the machine-learning model, which is implemented using the Python programming language. We utilized a dataset comprising approximately 1500 images, which includes a diverse group of individuals beyond just family and friends, to provide a balanced and initial evaluation of the application's accuracy.

3.2 Image Cleaning and Preparation

This phase is considered an important step to prepare the image for the next phases of dataset processing and modeling as well, which might be affected due to image quality issues. This phase mainly involved steps for cleaning the image.

1. *Initialize the Face Recognition Library:*

   By loading and configuring the face recognition library to prepare it for processing the images.

2. *Process each image:*

   Here, it iterates through each image in the dataset and uses the library to analyze the image for the presence of human faces.

3. *Verify face detection:*

   For each image, check the output of the face recognition library to determine if a face is detected.

4. *Filter Images:*

   In this step, retain only the images that contain human faces, and discard any images without detected faces.

5. *Proceed to the Next Phase:*

   At the end, pass the filtered set of images (those containing human faces) to the next phase for further processing or analysis.

3.3 Image Preprocessing

Preparing images is a crucial step in modeling, and it is necessary to address any issues with the image before applying machine-learning models. This ensures optimal results by utilizing clean and uniform data. Since image quality significantly influences the effectiveness of models, during preprocessing, the system determines the system determines the location and orientation of the face in the image. The face_recognition.face_locations(unknown_image)" function is used to determine the coordinates of the face. This step is important to standardize the input for the encoding phase, ensuring that the features are correctly extracted from the true regions of the image, so we can ensure that the next phase proceeds without errors.

3.4 Feature Selection

Feature selection, specifically feature engineering, plays a crucial role in the effectiveness of face recognition systems. This process involves identifying and extracting relevant features from images that can be used for accurate identification and verification of faces.

1. *Face Recognition Library:*

   The face recognition library, built on top of dlib, provides powerful tools for face detection and recognition. One of its key functionalities is the extraction of facial features using the face_recognition.face_encodings function.

2. Facial Feature Extraction:

The face_encognition.face_encodings function processes an image to identify and extract facial features. This includes detecting key facial landmarks such as the eyes, nose, mouth, and jawline, and calculating their relative positions. The library specifies 128 points on the faces of these landmarks. Once the locations of the eyes and mouth are known, the image is adjusted through simple transformations like rotation and scaling to center the eyes and mouth as accurately as possible, without introducing distortions. These transformations preserve parallel lines and are referred to as affine transformations.

3. Encoding Process:

The encoding relies on the distances between facial landmarks, which capture variations in facial structure and features while remaining resilient to changes in lighting, angle, and facial expressions.

The most straightforward method of face recognition involves directly comparing the unfamiliar face from Step 2 with all the images of people who have already been tagged. If we find a previously tagged face that closely resembles our unknown face, it is likely the same person.

4. Modeling:

The resulting encodings can be used in various machine-learning tasks, such as clustering similar faces, identifying individuals, or verifying identities against a database of known faces.

By transforming the raw image data into meaningful feature encodings, the model can efficiently compare faces and make accurate predictions. The next phase involves choosing the machine learning algorithms to be used.

3.5 Machine Learning Modeling

Machine learning falls under the umbrella of artificial intelligence and involves algorithms that enable systems to learn and recognize patterns. There are two main types of machine learning: supervised learning and unsupervised learning [33]. Supervised machine learning uses a predefined set of features to identify classification features and requires learning features from the input and output. With labeled training data, a supervised learning algorithm can learn from and make predictions about new data. On the other hand, unsupervised learning involves a learning system with input but no predicted output variables. Cluster analysis and association mining algorithms are examples of unsupervised learning approaches.

This section outlines the workflow of the proposed model and the machine learning algorithms that are at the core of the face recognition functionality.Fig.4 illustrates the workflow of the proposed convolutional network. Which is powered by the pre-trained models provided by the face recognition library. The encoded features from the known and unknown images from the previous step are compared using the "face_recognition.compare_faces function," which determines the probability of a match. Additionally, the "face_recognition.face_distance" function computes the Euclidean distances between face encodings to quantify the similarity between faces. These comparisons form the basis of this system, ensuring that the correct individuals are identified accurately.

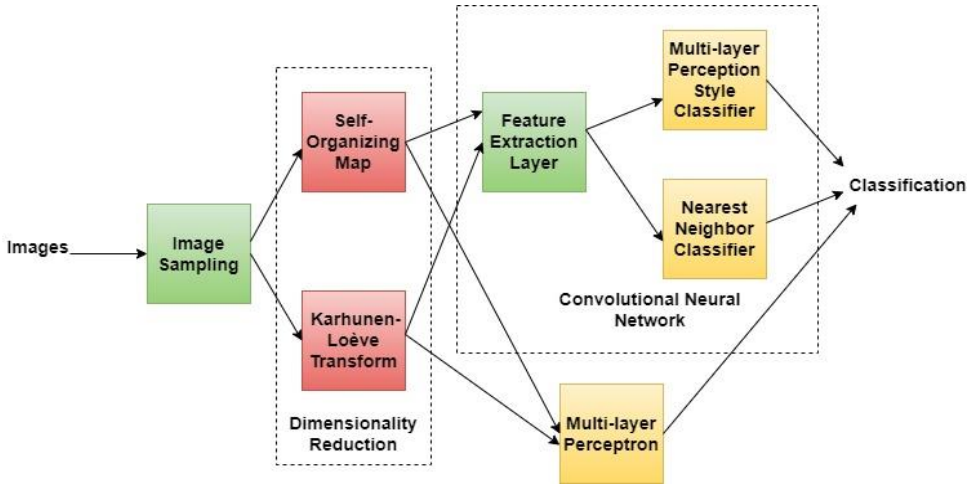

Fig. 4. Classification for face recognition using a convolutional network [34].

The conventional neural network is a subfield of machine learning. Fig.4 illustrates the structure of a Convolutional Neural Network, and how it works to classify the image and recognize the face. The neural network takes in data trained to recognize the pattern in this data and then predicts the outputs for a new set of similar data [31]. The steps below represent how it works in detail.

1. The image is represented as a pixel of an array and the Face landmark specifies 128 points in the face.

2. Send the input to each neuron, which is connected to the neurons of the next layer through a weight, and then provide the hidden layer neurons with the sum of these weights as input.

3. To this sum, add the bias and then run it through the activation function. Also, the neuron gets activated if the activation function gives a high value.

4. The method of operation is known as forward propagation. It calculates the error by comparing the projected output to the actual output.

5. Using backpropagation, change the amount of weight based on the error. Then Repeat the forward and backpropagation processes until the weights are allocated and the network can properly predict the form.

6. When the weights have been allocated, the training process is complete.

3.6 System Architecture

The user of the application will train their face to save it as a password for his chosen application that he wants to lock (File containing sensitive information). Each landmark that the system gets in his face will be the guide for the app to know if it is the user or not. Smartphone owner trains their face and save files on their smartphone, Fig.5 shows the architecture.

Fig. 5. System Architecture

The File Locker's conceptual structure is illustrated in Fig.6., The OpenCV and Dlib library are used to affect the facial recognition used to take the image. The image will be stored as a string or array. The detection of the face in pixels will be determined by the amount of facial area collected. This face image saved in a smartphone file will be used as a training dataset. It will be utilized as a baseline for analyzing and comparing the newly discovered and identified face images. The locked programs will only be unlocked if they match the stored facial image.

*The Dlib is a flexible and widely used facial recognition toolkit that strikes an optimal balance of resource utilization, accuracy, and latency for real-time face recognition in mobile app development. It's becoming a ubiquitous, if not essential, library in the facial recognition scene, and despite competition from more recent candidates, it's a solid candidate for your computer vision and facial identification or detection framework used with OpenCV to handle light conditions and rotate images.

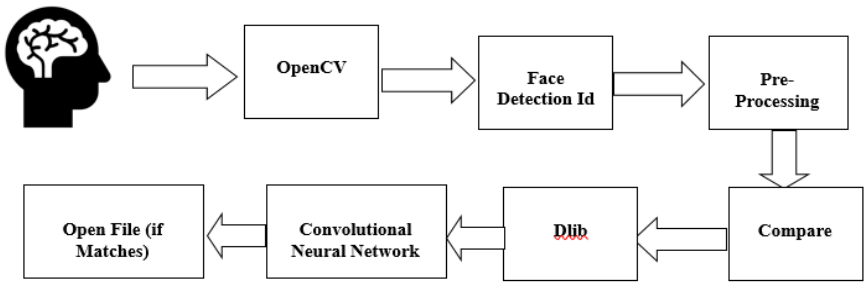

Fig. 6. Conceptual Framework

The app's interface is built using the Flutter framework, which enables cross-platform compatibility and provides a smooth, high-performance user experience. Flutter interacts seamlessly with backend services, handles data exchange, and manages interface updates in real-time. This framework was chosen for its flexibility and ability to work efficiently with Android operating systems, in line with the overall system architecture.

3.7 Automation

This phase integrates the face recognition process into the mobile application's workflow. The Flask web framework provides various endpoints such as /verify, /compare, and /recognition, enabling verification and recognition operations. These endpoints handle the entire process, from image capture to face comparison, ensuring that the application operates efficiently in real-time scenarios Fig.7 depicts the automation phases of the implemented Android application for securing files.

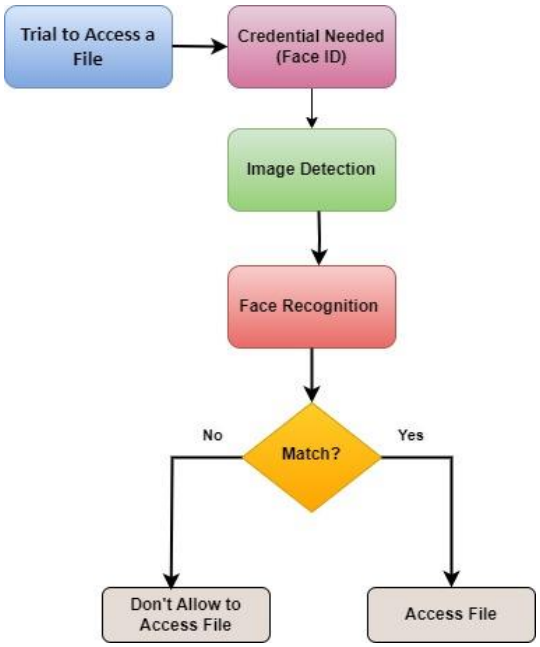

Fig. 7. Methodology used in this paper for access file.

When a person needs to access a file that contains sensitive information on the mobile device, the application lock requests a credential (face ID) then the model needs to detect a photo of the user's face who needs to open the file. This photo is then pre-processed and analyzed in the pre-processing stage it must be compared to the one stored in the file located on the server. This comparison is done using a specific machine-learning algorithm designed for facial recognition (CNN) and if it is the correct person opens the file, others reject the access file.

4. Result and Discussion

To evaluate the proposed application, you installed it on an Android mobile. The mobile application gives a menu for users to choose a file that needs to be secured.

The basic methods are face recognition and sometimes passwords. The first gives access to the phone's content by taking a picture of the user's face. If the user chooses the face recognition method, which is the most performant and very easy to use, then he/she will have to first take pictures of them. These photos will be stored in the database and used in the recognition process. The next step is setting a password for the situation where their face is not recognized, or to add a new face ID or remove it.

When the files are locked using this method, the user will have to scan their face to be provided access to the file.

We applied this application on Android version 10,11,12, and During testing, we observed consistent functionality and performance across these versions, with no significant variation in accuracy or speed as version 14 of Android.

Table 1. represents the specifications of one of the devices that were tested, we present this device considering it as the Android 14 version, which is the latest and most widely adopted version.

Table 1. Platforms used to make experiments**.**

Specification's name	**Specification Value**
Android Version	14
CPU	Octa-core (1x 3.00 GHz Cortex-X2,3x2.50GHz Cortex-A710, 4x 1.80 GHz Cortex-A510)
GPU	Qualcomm Adreno 730
RAM	12
ROM	256
Resolution	1440 x 3088 pixels
Self-camera	front camera has a 40 MP sensor with an f/2.2 aperture, 26mm (wide), and 1/2.82", 0.7 µm

4.1 The Application Interface

When a file is secured and the person tries to access it, the application shows a prompt on the screen asking the user to click to take a photo for face detection as shown in Fig.8.(A) and Fig.8.(B).

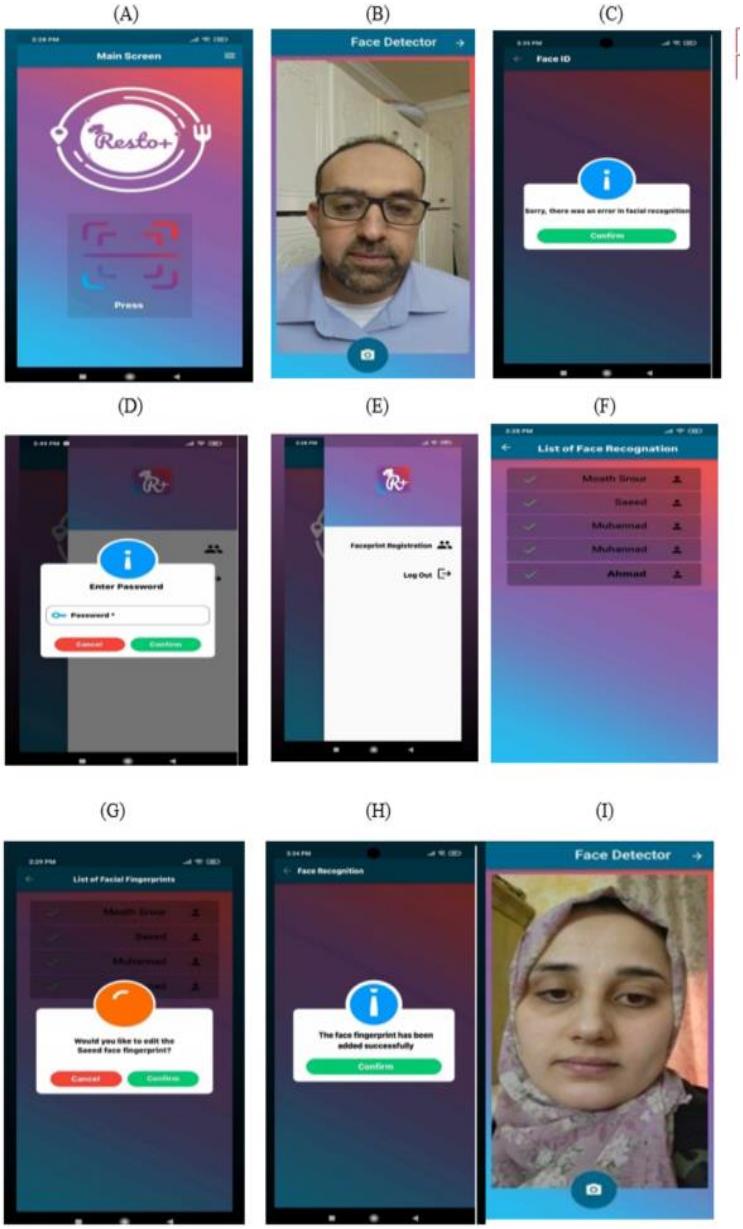

Fig. 8. Main screen (A), Face Detection (B), Message appears if the face is not correct, or does not exist (C). Enter passwords (D), Face ID Management (E), and User's Face ID, which are stored in a database (F). Edit face ID (G). Add face ID (H). Detect the face of women wearing shawl (I).

When a face does not match those in the database, it rejects access to the file and the message, which is shown in Fig.8.(C).

Now we returned to the main screen, exactly to appear of the main screen, where the admin can go to the add face management screen and register their faces, he has to enter his password again as shown in Fig.8.(D).

When the user needs to register a new Face ID, or add, or remove the Face ID, the application prompts them to enter the password as shown in Fig.8.(D). After entering the correct password, they can proceed with the Face ID management as shown in Fig.8.(E) and Fig.8.(F). If the user tries to edit the face ID, the application appears in the message shown in Fig.8.(G) and Fig.8.(H). If it is pressed confirm she can modify the face ID else return to the main screen of the list of facial fingerprints.

When the user is a woman, wearing a shawl as illustrated in Fig.8.(I), can detect the face and then recognize and compare it with a picture in the database and if it matches can access the file, if not the image is rejected.

Fig.9.(A) When a male participant tries to access the file with his eyeglasses, the application asks FaceId to register as illustrated in Fig.9.(B) After that, it recognizes the participant with eyeglasses and then he can access the mobile file, so when the participant wears eyeglasses the application recognizes his face and allows him to access for mobile file, also when he doesn't wear his eyeglasses as shown in Fig.9.(D) and try to access a mobile file, it is asked about his face ID and when the application recognizes his face it allows him to access the mobile file as illustrated in Fig.9.(F).

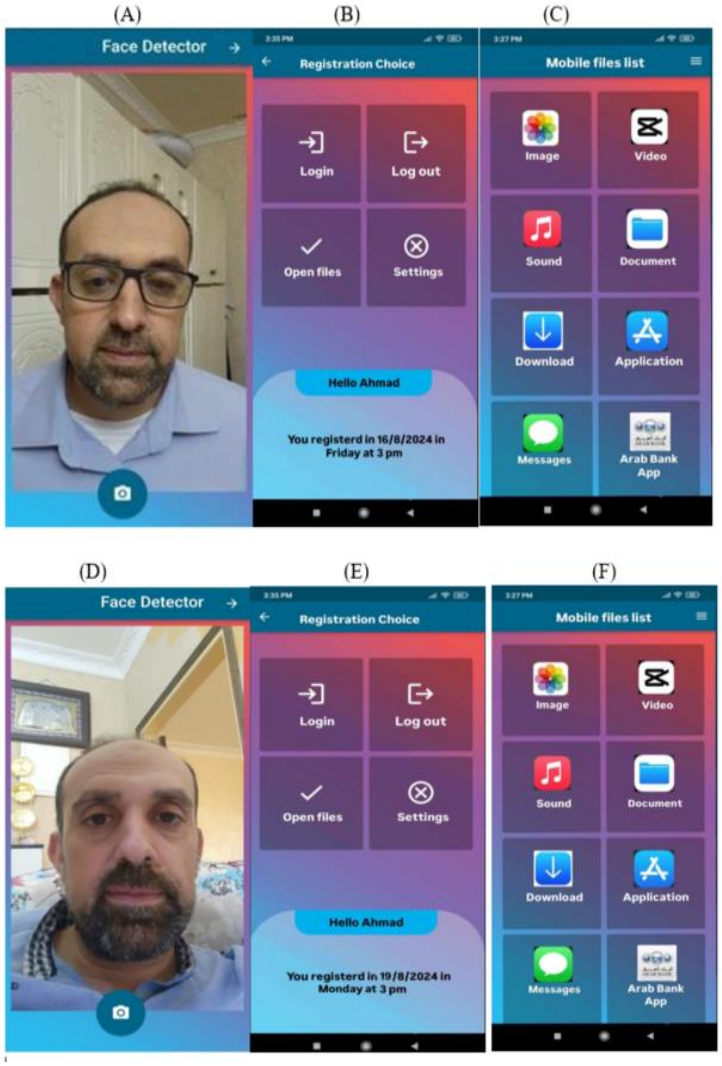

Fig. 9. Participant with eyeglasses (A), correct face ID and registered (B), list of choices on the mobile application (C). Participant without wearing his eyeglasses (D), correct face ID and registered (E), list of choices on the mobile application (F).

Table 2. Comparison between the proposed model in this paper and those of other studies in the literature.

Reference	Application	Accuracy	Long distance (m)	Performance	Face recognition used	Recognize Eyeglass and hat
[1]	App locker	93%	1.5	medium	yes	No
[21]	f-looker app	-	2	Very Satisfactory	yes	No
[24]	File management	91%	1.5	slow	yes	No
[25]	Applock with face	90%	1.5	slow	yes	No
[26]	Super app lock	85%	-	Too slow	yes	No
[27]	Smart lock app	-	2	Very slow	yes	No
-	Our application	99%	1	High	yes	yes

Fig. 10. Applock with face A [25], File management B [24], super app lock C [26], and smart lock app D [27].

When reviewing different applications, it is critical to examine both their advantages and disadvantages. Previous research has looked into how Android application lockers use biometric identification and user interface to protect phones. However, more approaches have flaws and may have difficulty recognizing people who wear eyeglasses or hats, particularly women who wear veils.

We investigate how machine-learning techniques can be utilized to overcome these constraints in facial recognition. Other research has looked at machine learning algorithms to detect faces and use in applications, as well as how many meters can capture a photo, such as the f-looker application, which takes only 2 m [25] App locker 1.5 m [1], but our application only takes 1m, which is one of the limitations of our application that we need to improve.

However, this study focuses further on the impact of the combination of facial recognition technology and a variety of characteristics in the context of Android lockers for file security, so it balances between the security of the file in the phone and efficiency.

While this study gives helpful insights into technical aspects and overall user views of mobile security applications, some applications continue to experience accuracy challenges. Our Android application, on the other hand, has a 99% accuracy rate, whereas App Locker has a 93% accuracy rate [1].

This study has also produced an easy-to-use application that is appropriate for all people, regardless of gender or age. However, there is still some misunderstanding over how although previous investigations have dealt with similar topics; they have not thoroughly investigated precisely the elements that this research is trying to find.

Fig.10. (A, B, C, D) shows the deed back from the Google store about each application and it shows below the details.

File management apps are shown in Fig.10.((B), particularly ones with broad functionality, may consume system resources and degrade device performance. This could be an issue with older or less powerful devices, Changes in Android versions can also affect the capabilities of file manager apps [24], and Smart Lock app [27] but in our application, we consider the performance of the application to be efficient, Applock with Face app [25]is good but it's a bit slow when it comes to scanning the face despite our application being scanned and Fig.10.(A) show the feedback, Super Applock with Face ID [26] the 6-digit PIN was invalid. The app only accepts four digits, and there is no way to delete the security after it has been added, it will only accept a passcode or a pattern. There is no facial unlock available, and we see the feedback for this Super Applock with Face ID in Fig.10. (C).

The Smart Lock app suddenly stopped working properly. The problem encountered is that it turns off and on, depending on which lock is used. However, if a checkmark to locked, it goes off every time someone exits compared with our application in this study it works faster and no problem happens when it runs the application Fig.10.(D) represents the feedback.

Finally, we believe that this study will provide important insights into the variables that impact the success of different safety measures, especially about individuals who use eyeglasses or headwear.

So, my Android app is used to secure sensitive files on smartphones. It adds application authentication using security measures and can be easily integrated into other systems. It is faster, easier, and more secure than passwords and PINs, and suitable for all ages. It is simple to use and suitable for people of all ages, including the elderly, children, and people with disabilities. It is also more convenient than utilizing difficult-to-remember passwords or PINs. Another benefit is that it is quite secure. It analyzes the distinctive features of the user's face, such as the distance between the eyes, the shape of the nose, and the contours of the face, using advanced algorithms. This makes it harder for someone to hack the phone with a bogus image. Overall, Face recognition is a dependable and safe method of protecting your phone and critical information. It is simpler to use, more accurate, and more secure than other applications. Face recognition is an excellent solution for everyone, whether they are elderly, children, or those with impairments.

5. Conclusion

The main contribution of this work is the safe file access solution it provides for Android. Modern security authentication methods are offered. Face recognition is fast and effective since it uses the CNNA algorithm from the OpenCV library, but it still depends on dlib. The developed Android application in this paper employs a variety of security safeguards to prevent unwanted access to applications or files holding sensitive information, such as email, social media, and chat passwords. The application provides advanced security by unlocking the phone with the owner's image, making it difficult for anybody else to access the phone using a phony image. The application is simple to use, appropriate for all ages, and more secure than passwords or PINs. The application is incredibly exact, identifying the owner's face with 99% percent accuracy. This capability can be added to the existing understanding of Android application lockers to avoid sensitive file hacking and close a research gap. We plan to incorporate liveness detection as a critical security feature. This functionality will detect real-time user presence by analyzing micro-movements like blinking or slight head motions, which are challenging to replicate with photos or masks. This approach minimizes the risk of unauthorized access through presentation attacks, ensuring that only real, live users can access sensitive files. Additionally, our future work will investigate adversarial training techniques to further secure the model against manipulation attempts specific to Android platforms, reinforcing the reliability of face-based authentication in real-world applications.

The creative process can be both challenging and rewarding. There are several stages and obstacles to overcome between the initial concept and the finished product. The app's ability to collect distances is less than 1 meter, which will be considered in future work. In addition to addressing difficult concerns such as headgear and spectacles to achieve high accuracy, as well as ambient aspects such as illumination.

References

[1] R. Stoleriu and M. Togan, "A Secure Screen and App Lock System for Android Smart Phones Using Face Recognition," *2020 13th International Conference on Communications, COMM 2020 - Proceedings*, pp. 133–138, Jun. 2020, doi: 10.1109/COMM48946.2020.9142008.

[2] "The Evolution of Smartphone Security." Accessed: Aug. 01, 2024. [Online]. Available: https://alltopstartups.com/2017/06/20/the-evolution-of-smartphone-security/

[3] M. R. Srilekha and M. D. Jayakumar, "A Secure Screen Lock System for Android Smart Phones using Accelerometer Sensor," *IJSTE-International Journal of Science Technology & Engineering*, vol. 1, no. 10, 2015, Accessed: Aug. 01, 2024. [Online]. Available: www.ijste.org

[4] "Android Statistics 2024 - By Market Share, Users and Revenue." Accessed: Aug. 01, 2024. [Online]. Available: https://www.enterpriseappstoday.com/stats/android-statistics.html

[5] M. Ali Shah, A. Khan, M. Ahmed, and S. Arshad, "Android Malware Detection & Protection: A Survey," *Article in International Journal of Advanced Computer Science and Applications*, vol. 7, no. 2, 2016, doi: 10.14569/IJACSA.2016.070262.

[6] V. Venkateswara Rao and A. S. N. Chakravarthy, "Analysis and bypassing of pattern lock in an android smartphone," *2016 IEEE International Conference on Computational Intelligence and Computing Research, ICCIC 2016*, May 2017, doi: 10.1109/ICCIC.2016.7919555.

[7] T. Saad Mohamed and T. Saad Mohamed Lecturer, "Security of Multifactor Authentication Model to Improve Authentication Systems Information and Knowledge Management Security of Multifactor Authentication Model to Improve Authentication Systems," vol. 4, no. 6, 2014, doi: 10.13140/RG.2.2.18515.53288.

[8] S. K. Datta, C. Bonnet, and N. Nikaein, "Android power management: Current and future trends," *2012 the 1st IEEE Workshop on Enabling Technologies for Smartphone and Internet of Things, ETSIoT 2012*, pp. 48–53, 2012, doi: 10.1109/ETSIOT.2012.6311253.

[9] M. Shah, J. D. Deng, and B. J. Woodford, "Video background modeling: Recent approaches, issues, and our proposed techniques," *Mach Vis Appl*, vol. 25, no. 5, pp. 1105–1119, Nov. 2014, doi: 10.1007/S00138-013-0552-7/METRICS.

[10] D. Van Bruggen, S. Liu, M. Kajzer, A. Striegel, C. R. Crowell, and J. D'Arcy, "Modifying smartphone user locking behavior," *SOUPS 2013 - Proceedings of the 9th Symposium on Usable Privacy and Security*, 2013, doi: 10.1145/2501604.2501614.

[11] D. Hintze *et al.*, "CORMORANT," *Proc ACM Interact Mob Wearable Ubiquitous Technol*, vol. 3, no. 3, pp. 1–23, Sep. 2019, doi: 10.1145/3351243.

[12] A. Bianchi, I. Oakley, V. Kostakos, and D. S. Kwon, "The phone lock: Audio and haptic shoulder-surfing resistant pin entry methods for mobile devices," *Proceedings of the 5th International Conference on Tangible Embedded and Embodied Interaction, TEI'11*, pp. 197–200, 2011, doi: 10.1145/1935701.1935740.

[13] "How to Protect Your Privacy on Android." Accessed: Aug. 02, 2024. [Online]. Available: https://spreadprivacy.com/android-privacy-tips/

[14] D. Huang, C. Shan, M. Ardabilian, Y. Wang, & L. Chen, "Local Binary Patterns and Its Application to Facial Image Analysis: A Survey". IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 4, issue 6, pp.765–78. Accessed: Aug. 02, 2024. [Online]. Available: https://sci-hub.ru/10.1109/TSMCC.2011.2118750

[15] X. Chao, G. Dave, and K. Sriadibhatla, "Face Recognition in Mobile Phones", Accessed: Aug. 02, 2024. [Online]. Available: https://www.researchgate.net/publication/228445783

[16] E. Vazquez-Fernandez, H. Garcia-Pardo, D. Gonzalez-Jimenez, and L. Perez-Freire, "Built-in face recognition for smart photo sharing in mobile devices," *Proc (IEEE Int Conf Multimed Expo)*, 2011, doi: 10.1109/ICME.2011.6012057.

[17] O. Dospinescu and I. Popa, "Face Detection and Face Recognition in Android Mobile Applications", doi: 10.12948/issn14531305/20.1.2016.02.

[18] S. Chaudhry and R. Chandra, "Design of a Mobile Face Recognition System for Visually Impaired Persons," Feb. 2015, Accessed: Aug. 02, 2024. [Online]. Available: https://arxiv.org/abs/1502.00756v2

[19] A. Salihbašic and T. Orehovacki, "Development of an android application for gender, age and face recognition using OpenCV," *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2019 - Proceedings*, pp. 1635–1640, May 2019, doi: 10.23919/MIPRO.2019.8756700.

[20] M. Harbach, A. De Luca, and S. Egelman, "The anatomy of smartphone unlocking a field study of android lock screens," *Conference on Human Factors in Computing Systems - Proceedings*, pp. 4806–4817, May 2016, doi: 10.1145/2858036.2858267.

[21] A. L. A. Ramos, M. A. M. Anasao, D. B. Mercado, J. A. Villanueva, C. J. A. Ramos, A. A. T. Lara, &C. N. A. Margelino, "*F-Locker: An Android Face Recognition Applocker Using Local Binary Pattern Histogram Algorithm*". International Journal of Computer Science and Information Security (IJCSIS), vol. 16, issue 2. pp. 129-135, 2018.

[22] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1635–1650, Jun. 2010, doi: 10.1109/TIP.2010.2042645.

[23] R. Sutoyo, J. Harefa, Alexander, and A. Chowanda, "Unlock Screen Application Design Using Face Expression on Android Smartphone," *MATEC Web of Conferences*, vol. 54, p. 05001, Apr. 2016, doi: 10.1051/MATECCONF/20165405001.

[24] "File Manager Plus." Accessed: Aug. 02, 2024. [Online]. Available: https://www.alphainventor.com/file-manager-plus

[25] "Applock with Face – Apps on Google Play." Accessed: Aug. 02, 2024. [Online]. Available: https://play.google.com/store/apps/details?id=com.sm.faceapplock

[26] "SuperX AppLock with Face ID – Apps on Google Play." Accessed: Aug. 02, 2024. [Online]. Available: https://play.google.com/store/apps/details?id=com.gmcnt2.superlockapp
[27] "Smart Lock (App/Photo) – Apps on Google Play." Accessed: Aug. 02, 2024. [Online]. Available: https://play.google.com/store/apps/details?id=ukzzang.android.app.protectorlite
[28] "[2212.13038] A Survey of Face Recognition." Accessed: Nov. 07, 2024. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2212.13038
[29] I. Adjabi, A. Ouahabi, A. Benzaoui, and A. Taleb-Ahmed, "Past, Present, and Future of Face Recognition: A Review," *Electronics 2020, Vol. 9, Page 1188*, vol. 9, no. 8, p. 1188, Jul. 2020, doi: 10.3390/ELECTRONICS9081188.
[30] M. Wang, W. D.- Neurocomputing, and undefined 2021, "Deep face recognition: A survey," *ElsevierM Wang, W DengNeurocomputing, 2021•Elsevier*, Accessed: Nov. 07, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231220316945
[31] K. He, X. Zhang, S. Ren, J. S.-P. of the IEEE, and undefined 2016, "Deep residual learning for image recognition," *openaccess.thecvf.comK He, X Zhang, S Ren, J SunProceedings of the IEEE conference on computer vision and, 2016•openaccess.thecvf.com*, Accessed: Nov. 07, 2024. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html
[32] Y. Liu, J. Stehouwer, A. Jourabloo, and X. Liu, "Deep tree learning for zero-shot face anti-spoofing," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 4675–4684, Jun. 2019, doi: 10.1109/CVPR.2019.00481.0.
[33] N. Sharma and I. Ghosal, "How Machine Learning Algorithms Work in Face Recognition System? A Review and Comparative Study," *International Journal for Innovative Engineering & Management Research*, vol. 12, issue. 3, pp. 250-262, 2023. [Online]. Available: https://papers.ssrn.com/abstract=4397531
[34] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Trans Neural Netw*, vol. 8, no. 1, pp. 98–113, 1997, doi: 10.1109/72.554195.

Authors' Profiles

Marah Radi Hawa works as a Computer Engineer in the public sector in Palestine, where I began my professional career in 2017. I earned my Bachelor's degree in Computer Systems Engineering in 2016. Currently, I am pursuing a master's degree in cyber security at the Arab American University in Palestine.

Amani Yousef Owda Assistant Professor in Computer Engineering and Data Science in the Faculty of Graduate Studies at the Arab American University in Palestine. She worked as a head of the Department of Natural, Engineering, and Technology Sciences in the Faculty of Graduate Studies at Arab American University from 2022 -2023. She worked as a research associate in the Faculty of Engineering at the University of Manchester from 2019 - 2020. In addition, she worked in the School of Engineering at Manchester Metropolitan University from 2015 to 2019. She worked at Birzeit University from 2007- 2011. She received her MSc. degree (Hons.) from The University of Manchester, UK in 2013, and her Ph.D. degree in Computer Engineering from Manchester Metropolitan University, UK in 2018. Dr Owda won the Arab American University Award for Excellence in Scientific Research in 2023 and the Best Paper award in SPIE Europe Security+Defence, Warsaw, 2017. Since 2018, she has led research in multi-disciplinary fields with a focus on artificial intelligence, machine learning, decision support systems, image processing, medical applications of microwave and millimeter-wave imaging, security screening, and anomaly detection. She has published more than 53 articles in well reputable journals. She is a reviewer in many well-known Journals, and she is supervising MSc and PhD students.

Majdi Owda Associate Professor in Computer Science and Dean of Faculty in Artificial Intelligence and Data Science at the Arab American University in Palestine. In addition, he is a UNESCO Chair for Data Science for Sustainable Development. He worked as a head of the Department of Natural, Engineering, and Technology Sciences in the Faculty of Graduate Studies at Arab American University from 2020 -2022. I worked at the School of Computing, Mathematics, and Digital Technology at Manchester Metropolitan University from 2009 to 2020. He gained a BSc in Computer Science from the Arab American University in 2004, an MSc in research in Computer Science with distinction from Manchester Metropolitan University in 2005, and a Ph.D. in Computer Science in 2011. His main research interests are AI Techniques for Natural Language Interfaces to Relational Databases, Data Science, Conversational Informatics, Conversational Agents, Knowledge Trees, Knowledge Engineering, Planning, Information Extraction, AI Techniques for the Help of Society, Web/Data/Text Mining, Digital Forensics Processes and Frameworks, Digital Forensics Artefacts, Information Retrieval from Large Data Sources, Internet of Things Frameworks and Internet of Things Digital Forensics Artefacts and Security.

How to cite this paper: Marah Radi Hawa, Amani Yousef Owda, Majdi Owda, "Android Mobile Security and File Protection Using Face Recognition", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.15, No.2, pp. 26-40, 2025. DOI:10.5815/ijwmt.2025.02.03