

Article

Tracking Unauthorized Access Using Machine Learning and PCA for Face Recognition Developments

Vasile-Daniel Păvăloaia *  and George Husac

Department of Accounting, Business Information Systems and Statistics, Faculty of Economics and Business Administration, Alexandru Ioan Cuza University of Iasi, 700506 Iasi, Romania

* Correspondence: danpav@uaic.ro

Abstract: In the last two decades there has been obtained tremendous improvements in the field of artificial intelligence (AI) especially in the sector of face/facial recognition (FR). Over the years, the world obtained remarkable progress in the technology that enhanced the face detection techniques use on common PCs and smartphones. Moreover, the steadily progress of programming languages, libraries, frameworks, and tools combined with the great passion of developers and researchers worldwide contribute substantially to open-source AI materials that produced machine learning (ML) algorithms available to any scholar with the will to build the software of tomorrow. The study aims to analyze the specialized literature starting from the first prototype delivered by Cambridge University until the most recent discoveries in FR. The purpose is to identify the most proficient algorithms, and the existing gap in the specialized literature. The research builds a FR application based on simplicity and efficiency of code that facilitates a person's face detection using a real time photo and validate the access by querying a given database. The paper brings contribution to the field throughout the literature review analysis as well as by the customized code in Phyton, using ML with Principal Component Analysis (PCA), AdaBoost and MySQL for a myriad of application's development in a variety of domains.

Keywords: face detection; machine learning algorithms; principal component analysis; AdaBoost



Citation: Păvăloaia, V.-D.; Husac, G. Tracking Unauthorized Access Using Machine Learning and PCA for Face Recognition Developments.

Information **2023**, *14*, 25. <https://doi.org/10.3390/info14010025>

Academic Editors: Eftim Zdravevski, Petre Lameski and Ivan Miguel Pires

Received: 20 September 2022

Revised: 18 December 2022

Accepted: 27 December 2022

Published: 30 December 2022



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Individual survival in a socially complex environment is heavily reliant on the capacity to understand visual information about a person's age, gender, ethnicity, identity, and emotional state based on that person's face. Despite a range of challenging settings (numerous facial expressions and postures, alterations in light and appearance), human beings can execute face identification with astonishing consistency without conscious effort. FR problem is considered one of the most proficient and profitable application of ML and computer vision [1]. Although FR research utilizing automated or semi-automatic algorithms began in the 1960s [2], has gotten considerable interest in the last 20 years. FR algorithms have a wide range of conceivable applications, which is one reason for its current rising popularity. Another factor is the widespread availability of inexpensive hardware, including digital cameras and video cameras, which has made capturing high-quality, high-resolution photographs in a considerably facile manner. Despite the increased interest, existing state-of-the-art FR algorithms function effectively when facial photographs are taken in consistent and controlled situations. FR systems that perform reliably in uncontrolled conditions, on the other hand, are still a topic of research.

Even though there are a variety of viable biometric techniques that work well today, such as fingerprint analysis as well as iris scans, these techniques require person's participation and adhere to a rather rigorous data collecting process. FR allows for greater flexibility because participants are not needed to collaborate or even be aware that they are being examined and recognized. As a result, FR is a less invasive and perhaps more

successful identifying tool. With the advancement of information technology, the desire for an accurate personal identification system based on detecting biological traits is rising, rather than traditional systems that employ ID cards or PINs. The face is the most recognized and identifiable of all bodily features, therefore utilizing it for identification reduces the necessity for direct contact, as well as any psychological or physical opposition, such as that faced while trying to acquire fingerprints.

Within the current research, in the next subchapters, will be performed a literature review that analyzes the body of literature published, starting from the first prototype delivered by Cambridge University [3,4] until the most recent discoveries for identifying the ML algorithms for FR as well as the programming languages and environments that are mostly used (Sections 1, 1.1 and 1.2).

Furthermore, the research aims to build a FR application that facilitates a person's face detection using a photo and validate access by querying a given database. The applicability is immense as it can be integrated in any domain that requires facial identification. While similar applications may require purchase and maintenance, the source of the current one is free and available online: <https://github.com/GeorgeHg98/Facial-Recognition-Software> (accessed on 8 December 2022).

This article's guiding concepts are simplicity and efficiency. Among others, the research contributes through a clear and succinct guideline on how to build a FR system. It also employs generally available and free development technologies, while the hardware requirements are minimal. This is making it an ideal alternative for academics and/or developers interested in this area. Thus, the solution brings a less costly, efficient, and secure way of authorizing employees' access (Section 2).

The research emphasizes (in Sections 3 and 4) that, notwithstanding the ethical concerns associated with the widespread use of AI technology, FR systems' capabilities are limitless, and their potential is genuinely immense.

1.1. Literature Review on ML Algorithms for FR

In order to investigate on the existing knowledge, studies, and debates relevant to FR research, as well as the architecture, technology, applications, and the limitations of the software it has been pursued a literature review within the latest research using the Clarivate Web of Science database of articles.

The main research questions that we have addressed throughout the literature review, are:

RQ1. What are technologies (algorithms and programming environment) mostly used for developing a FR application?

RQ2. What are the areas where FR applications are mostly seek?

1.1.1. Methodology Approach

For the purposes of undertaking the literature analysis, the Clarivate Web of Science was used, and the period was from anytime until the current year. The keywords search within Topic (TS), Title (TI), and Abstract (AB) returned 61 articles (Figure 1). After filtering the results (open access-only) a total of 44 reliable results were retained.



Figure 1. The keywords used on Web of Knowledge database platform.

tion accuracy, increasing resilience to expressions, and, more recently, enhancing algorithm effectiveness [1,9–11].

Majority of approaches [1] in FR rely upon Principal Component Analysis (PCA), including [11,12] who investigated the capabilities and limitations of PCA [13] by adjusting the number of eigenvectors and the size of range pictures [14–16] use PCA to create a new mapping of 3-D information to a spectrum, or depth, picture and also to partition the face into sub-regions that use the nose as an anchor, PCA to minimize feature space dimensionality, and the shortest distance for matching. Another important research trend is focused on the Iterative Closest Point (ICP) method, which has been used in a variety of ways for 3D form alignment, matching, or both. [17] introduced the very first insight into this type of approach to FR, after which [18] created an extended model to adapt to expressive variations, and [19] recommended to implement ICP to a set of relevant sub-areas rather than the entire face. Since a genuine face 3D form and texture completely represents it, we can consider appropriate to use both types of data (geometry and color or intensity) to boost identification accuracy: Multi-Modal (3D + 2D) biometric authentication is based on this principle.

The academic research conducted by [20] focuses on PCA to evaluate the picture's range and the intensity/color of the picture to the gallery. [21] introduced a four-dimensional (4-D) registration approach based on Iterative Closest Point (ICP) and texture while [22] suggest Eigen decomposition of flattened textures and canonical pictures for multi-modal 3D + 2D identification. Other researchers combine 3D and 2D resemblance scores derived from matching 3D and 2D profiles [23] or derive a feature representation based on Gabor filter responses in 2D and point signatures in 3D [24,25].

More recent attention has focused on the use of PCA but in combination with Python (and less with C++) programming language, as the PCA package is widely available, we will further present the concept of PCA and the mathematical basis of it that mainly relies on eigenvectors, eigenvalue, and eigenfaces.

1.1.2. PCA

In 2002, in the beginning of FR era, PCA was referred to by [26] as the karhunen-loeve transformation. The same authors [26] and others [27,28] initially defined it as the conventional method for data reduction and extraction of features in statistical pattern recognition and signal processing. Since the pattern frequently contains redundant data, mapping it to a feature vector can eliminate this redundancy while preserving most of the pattern's intrinsic relevant information [29]. These selected features play a key role in differentiating input patterns. Figure 3 consists in a visual representation of the PCA approach for FR model.

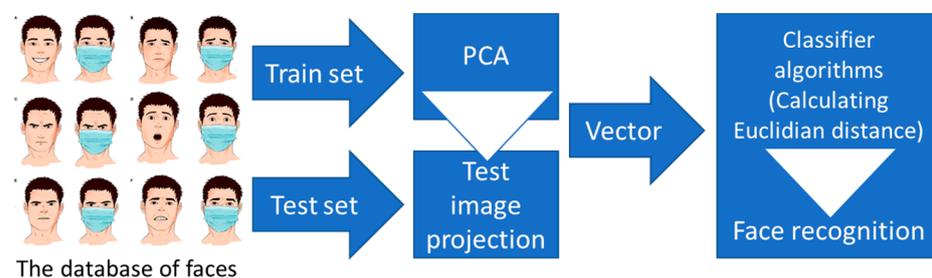


Figure 3. PCA approach for FR development.

A 2-D face image of size $N \times N$ can also be thought of as a one-dimensional vector of dimension N^2 . For instance, a face image from the ORL (Olivetti Research Labs) repository with the size 112×92 could be thought of as a vector with the dimension 10,304, or a point in a 10,304-dimensional space. A picture ensemble corresponds to a set of points in this vast space. Face images, because of their pretty much identical arrangement, will not be distributed randomly in this massive image space and can therefore be characterized by a

relatively low dimensional subspace. The fundamental concept behind the PCA is to locate the vectors that effectively account for the distribution of face pictures across the full image space. These vectors constitute the “facial space”, and it is the subspace of pictures of faces. Every one of those vectors is N^2 in length, represents an $N \times N$ picture, and stands for a linear model of the original facial images. Researchers call such vectors “eigenfaces” since they are the eigenvectors of the covariance matrix associated with the original face pictures and have a face-like look [29,30].

The related eigenvalues facilitate the prioritization of the eigenvectors based on their utility in defining picture variance. The eigenface pictures generated by L 's eigenvectors span a basis set that may be utilized to characterize the facial image. [31–34] assessed a restricted implementation of this architecture on 115 pictures ($M = 115$) of Caucasian men processed in a controlled manner and discovered that 40 eigenfaces ($M' = 40$) were adequate for a very acceptable description of the facial image. Consequently, a smaller M' can suffice for recognition because a perfect reconstruction of the picture is not required. The procedure is a pattern-matching task rather than an image reconstruction challenge in the context of FR software. The eigenfaces cover an M' dimensional subspace of the initial N^2 image space, and therefore the M' significant eigenvectors of the L matrix with the biggest corresponding eigenvalues are adequate for trustworthy representation of the faces in the eigenfaces' face space.

Our Database of Faces (ORL) [35] will be used as a source of images for the following example because it was developed and successfully applied by the Speech, Vision, and Robotics group of Cambridge University [4] in one of the most renowned and respected FR software projects ever made. In addition, this project was one of the first face detection research study of this dimension and one of the pioneers for this technology.

1.1.3. AdaBoost

This academic paper will make use of Python programming language with the Open-Source Computer Vision Library (OpenCV) for developing a FR application. Furthermore, the FR package, the ML algorithms and FR features of Python rely on AdaBoost classifiers [36]. From the above reason, the AdaBoost algorithm will be briefly presented in the next paragraphs.

The main concept of AdaBoost resides in the fact that from the same training set there will be produced two categories of classifiers [36,37]: weak and strong, and then they will be combined. This approach is implemented by altering data distribution. Each sample weight is validated based on whether the categorization of each sample in the training set is correct, as well as whether or not the previous overall classification was correct. The next classifier will train the new dataset and ultimately, all the classifiers will eventually be merged to form a final classification conclusion [38].

AdaBoost's multiple training images are performed by altering all sample weights [3,39,40]. Originally, each sample is assigned the same weight (the weight indicates the probability that the sample will be accepted into the testing phase by the classifier), and training a weak classifier is dependent on this. If the sample is successfully categorized, the weight is decreased. Alternatively, the weight is enhanced. As a result, the incorrectly classified sample will be noted, resulting in a new sample set. The next weak classifier in row, will be constructed by training the new sample set, which will adjust the weight. In an iterative loop, for example, it can obtain a certain number of weak classifiers.

Finally, the strong classifier will be constructed by overlaying the weak classifiers based on a predetermined weight. In order to use the algorithm of AdaBoost [41,42] it assigns a provided training $S = \{(x_i, y_i) \mid i=1, 2, \dots, n\}$, $x_i \in X$, $y_i \in Y$, wherein X is sample description while Y is sample representation. $y_i \in (0,1)$ in FR using this algorithm represents that 1 denotes a face while 0 denotes a non-face [43].

1.2. Current Study

Together, the studies considered indicate that AdaBoost algorithm is regarded as the most efficient [44] and reliable [45,46] for the current development purpose. Consequently, the FR development in this study will employ the AdaBoost technique since, according to the literature review, it was deemed essential for the creation of a solid FR application.

The paper contains customized code in Python, using ML with PCA and MySQL with myriad of applications in multiple domains (security, financial, etc.) and illustrates the development steps for a FR module that can be integrated in a wide variety of applications for plenty domains.

This FR system's compactness makes it extremely simple to build and modify. It takes two steps to add a new individual to the database, and the program recognizes it instantly. Furthermore, in contrast with previous studies that simply give theoretical explications of their approach, the authors made the program available for anybody to use, develop, and further investigate this issue.

2. Materials and Methods

There is a growing body of literature that recognizes the importance of FR algorithms for the development of a trustworthy face detection application in a myriad of domains. Thus, the authors contribute to the literature in the domain, by building a FR module to fulfil the gap identified in the literature review, namely the financial domain lacks such developments.

The proposed system consists in a live detection of a person, matching the face with the entries in a database. In addition to the code customization that was developed in Python, this practical study brings some contributions to the literature in the domain as it implements a skin tone identification feature for the advanced database search optimizations. Moreover, the development can be used as an input module for any (sector) application, not just financial, that requires user authentication based on face image recognition.

In the current and following Section 3 (Results) will be illustrated the development phases for the proposed FR module. The dataset used was Our Database of Faces [3] and was obtained freely from AT&T Laboratories Cambridge [35].

The architecture of the proposed FR system developed in this paper is illustrated in Figure 4. Firstly, the sign-up in the application is working similar to any modern mobile banking app, where the user adds an existing photo of himself or even take one, instantly. Most applications ask the user to set an account and a picture is loaded in a special database for faces. Later, the information provided by the client such as first name, last name, address, job, and birthplace are loaded in the MySQL Workbench database which keeps all the user's information, together.

After the face is loaded into the Pickle Database, it is analyzed using FR technology and algorithms from Python packages. Furthermore, an ONNX file loads an improved ML and DL model for analyzing the face as explained in the literature review section. Once the face is analyzed, the points of the face that have an outmost importance, usually the ones that do not change, are transformed into eigenvalues, and stored in an array. At this point, everything is set for the user and every time the login function is used, the software will take a live picture of the person that sits in front of the camera or passes by. After the picture is taken, the system will analyze and interpret the face from the photo by using ML. Furthermore, it will crop the face, eliminating the background and other irrelevant information from the picture. After this step, the application will map the eigenvalues of the user and try to match it with the existing information from the database. If any of the faces in the database match the one that is currently being used, this step is verified. Additionally, the system will check that the name and face match so that it can prevent other database users from accessing this user's content. After this phase confirms, the user has complete access to the program and all necessary application's features.

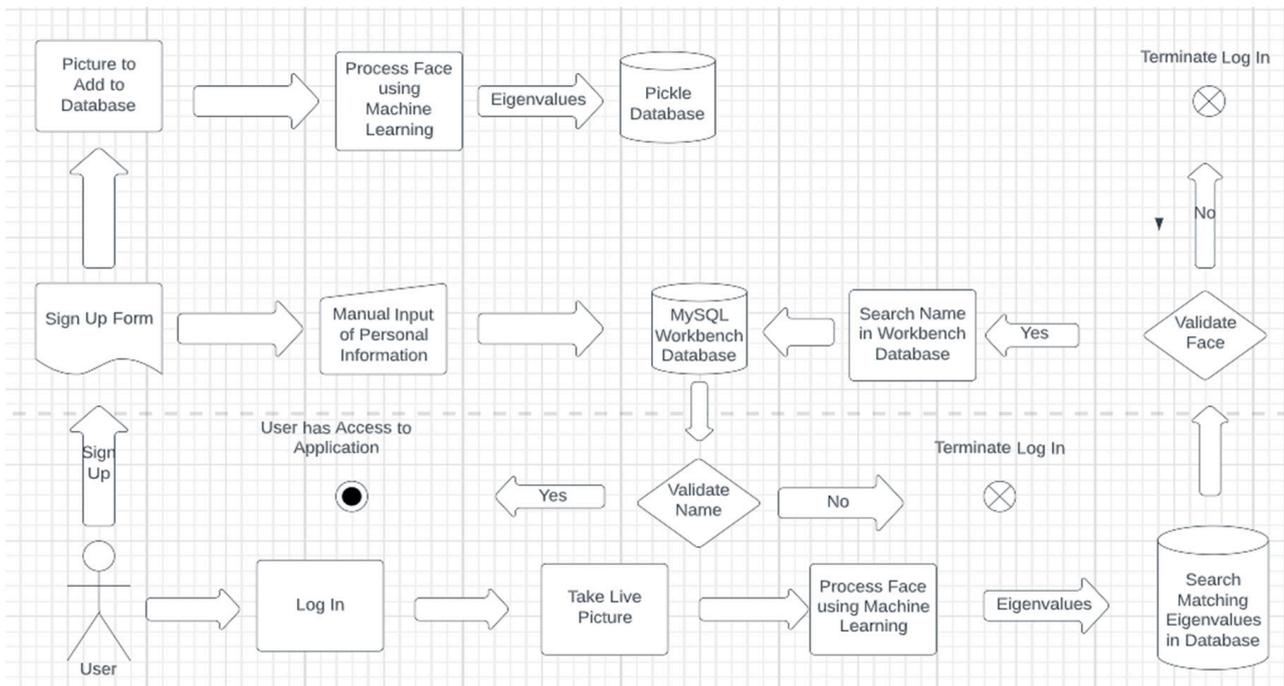


Figure 4. Architecture of the developed FR Software.

Face verification or face identification are the two general purposes for which FR systems are developed [47]. Verification, often referred to as 1:1 matching, is used to authenticate a person's true identification. Since the person is typically aware that is being scanned and may position properly to ensure that the camera can clearly see his face, FR algorithms typically have excellent accuracy on verification tasks approaching 99%. The challenge arrives for the identification case. The process of identifying someone is known as 1:N or 1:many matchings, and it involves using software to compare an unknown face to a database of known faces. Only under perfect circumstances, with consistent lighting and location, as well as when the participants' face features are visible and not covered, is the degree of precision achievable. Accuracy rates are often far lower in real-world installations. In this respect, we elaborated a chronological analysis of accuracy, and the results are included in Table 1. The purpose is to place the current development accuracy among the existing research.

In the current research, the authors evaluated the algorithm on a small group of persons. The environment was stable, but the algorithm was also checked when the environment alternates and the person (to be recognized) does not look at the camera or has his face tilted in other direction, simulating an involuntary photo. The algorithm proved to recognize the face even in the case of the involuntary photo, maintaining the accuracy of 96%. There have been made multiple tests for each person and the accuracy remained consistent at 96%.

The authors reveal that during the tests, there were few instances when the program failed to identify the face, although the subject was in a stable setting and was looking directly at the camera. This indicates that it was a software error not an algorithm issue. These are the cases that conducted to lowering the accuracy to 96%.

The CSIS [47] test, which was conducted at an airport's boarding gates, is extremely comparable to the test conducted by the authors of this study since it compares a live snapshot with the photo bank. The best FR algorithm in terms of accuracy, according to CSIS's report, obtained a score of 94.4% and this implementation benefited from a variety of FR algorithms. The present algorithm has a higher accuracy than CSIS's where the technology of airport cameras is significantly more sophisticated than the ones used for this research. Thus, the authors consider that the existing algorithm can operate more

efficiently and accurately with more sophisticated hardware based on the aforementioned reference [47].

Table 1. A comparison of algorithms accuracy for FR.

Researchers	Year	Dataset(s)	AI Technique	Accuracy
Yin et al. [48]	2017	NSL-KDD	Recurrent neural network	83.28% (binary), 81.29% (multiclass)
Jia et al. [49]	2019	KDD Cup 99 and NSL-KDD	Deep Neural Network	98%
Vinayakumar et al. [50]	2019	KDD Cup 99, NSL-KDD, Kyoto, UNSW-NB15, WSN-DS and CICIDS 2017	Deep neural network	Big variations between datasets
Kasongo et al. [51]	2019	NSL-KDD	Deep neural network	86.76% (binary), 86.62% (multiclass)
Kanimozhi et al. [52]	2019	UNSW-NB15	Deep neural network	89%
Mahalakshmi et al. [53]	2021	UNSW-NB15	Convolutional neural network	93.5%
Fu et al. [54]	2022	NSL-KDD	Deep neural network	90.73%
Mijalkovic J., et al. [55]	2022	NSL-KDD and UNSW-NB15	Deep neural network	97% for UNSW-NB15

Since the confidence interval for this study was set at 0.3, which is relatively small, the algorithm has a high security rate, which was further demonstrated by the fact that there were never encountered instances of false negative results. This suggests that the application never recognizes a different individual or grants access to a user who is not authorized.

Regarding the accuracy analysis in Table 1, every research study that was conducted evaluated on datasets that compare stable images rather than live images with images from databases. The current model proposes a real time FR system, and it is not built for photo comparison. The authors believe that the proposed model will be at least as competitive in matter of accuracy with the other algorithms that score higher [49,55] if it will be programmed to analyze datasets.

3. Results

The developed application was written in Python programming language and the code editor chosen is Visual Studio Code. Python is a general-purpose [56], high-level, interpreted programming language. Its design concept prioritizes code readability by employing heavy indentation. Python is garbage-collected and flexibly typed. It covers a wide variety of computing paradigms, including organized (especially procedural), object-oriented, and functional programming. Due to its extensive standard library, it is frequently referred to as a “batteries included” language. Python is routinely ranked among the top programming languages and because of the growing popularity of ML and AI an increased number of authors and software developers name it the programming language of the future.

Visual Studio Code, a.k.a. VS Code, is a source-code editor developed by Microsoft for Windows, Linux, and macOS. It provides support for debugging, syntax highlighting,

intelligent code completion, snippets, code refactoring, and integrated Git are among the features.

The Python version used in this manuscript is Python 3.9.7 and the package installer used is Pip 21.2.3. The packages installed through pip in the computer's environment are the following (Figure 5):

- OpenCV—a programming function library targeted mostly at real-time computer vision, this library uses ML algorithms to search for a face in the given picture;
- NumPy—a Python library that adds support for big, multi-dimensional arrays and matrices, as well as a wide variety of high-level mathematical functions to operate on these arrays;
- MySQL Connector—a Python library that provides interaction between the MySQL Workbench Database and the application;
- Pickle—a library that serializes and deserializes Python object structures. It is used in this material to serialize the array of face coordinates for the database of faces loaded in the application;
- Open Neural Network Exchange (ONNX) library—an open-source AI ecosystem comprising of technology businesses and academic groups that build open standards for describing ML techniques and software tools to foster AI innovation and cooperation.
- ONNX Runtime—a cross-platform network accelerator for DL with a versatile interface for integrating hardware-specific libraries.

```
from xml.dom import NotFoundErr
import cv2
import numpy as np
import k_means as km
import mysql.connector
import pickle
import face_recognition as fr
import onnx
import onnxruntime
from onnx_tf.backend import prepare
```

Figure 5. Python libraries used.

The database used for this development is a MySQL Workbench database (Figure 6). MySQL Workbench is a visual database design system that involves SQL programming, administration, database design, construction, and maintenance for the MySQL database system in a single integrated working environment. The database is loaded with 5726 different persons and their information (details are displayed on Figure 6).

1 • SELECT * FROM persons.persons;

firstname	lastname	age	skin_tone	job	birth_place	address
Jason	Petty	49	light	Barrister's clerk	Fete	1300 Gilmer Ave, Tallassee AL 36078
Teri	Garr	38	light	Amenity horticulturist	Breaza	890 Odum Road, Gardendale AL 35071
Rafael	Vinoly	47	light	Public affairs consultant (research)	Covasna	6140A Univ Drive, Huntsville AL 35806
Jason	Petty	38	light	Museum/gallery exhibition officer	Bragadiru	7855 Moffett Rd, Semmes AL 36575
Teri	Garr	58	light	Personal assistant	Tulcea	2900 S Mem PkwyDrake Ave, Huntsville AL 35801
Rafael	Vinoly	31	light	Homeless support worker	Solca	70 Pleasant Valley Street, Methuen MA 1844
Joey	Mantia	18	light	Illustrator	Ovidiu	350 E Fairmount Ave, Lakewood NY 14750
Laura	Bush	27	light	Secretary	Uricani	450 Highland Ave, Salem MA 1970
Didier	Defago	37	light	Biotechnologist	VaE	780 Lynnway, Lynn MA 1905
Roger	King	45	light	Marketing manager (social media)	Bedlean	67 Newton Rd, Danbury CT 6810
Catherine	Woodard	49	light	Clinical biochemist	Reghin	655 Boston Post Rd, Old Saybrook CT 6475
Paul	Wals	27	light	Nanoscientist	Podu Iloaiei	3300 South Oates Street, Dothan AL 36301
Janet	Leigh	41	light	Lecturer (further education)	Huedin	1818 State Route 3, Fulton NY 13069
Vladimir	Putin	46	light	Debt/finance adviser	Bolde	145 Kelley Blvd, Millbrook AL 36054
Jean	Carnahan	36	light	Human resources officer	Brad	6140A Univ Drive, Huntsville AL 35806
Leadev	Flood	43	light	Commercial/residential rural survey	Pandru	506 State Road, North Dartmouth MA 2747

Figure 6. MySQL Workbench Database.

The application consists of six Python files that contain different classes and methods that made the FR possible, an ONNX file which loads the ML algorithms, other files responsible for loading data in the database, and pictures used in the process. An overview of these can be depicted from Figure 7.

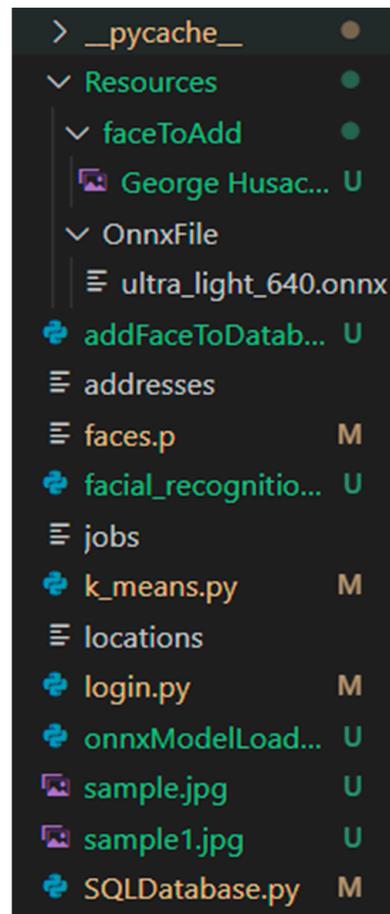


Figure 7. An overview of application files.

Firstly, the ONNX File `ultra_light_640.onnx` was added into the application for the application to have the trained ML algorithm available. Secondly, the model was loaded by executing the file `onnxModelLoader`, and the loader can be seen in Figure 8.

```
1 import onnx
2 import onnxoptimizer
3
4 onnx_model = onnx.load("Resources/OnnxFile/ultra_light_640.onnx")
5 passes = ["extract_constant_to_initializer", "eliminate_unused_initializer"]
6 optimized_model = onnxoptimizer.optimize(onnx_model, passes)
7
8 onnx.save(optimized_model, "Resources/OnnxFile/ultra_light_640.onnx")
9
```

Figure 8. Code of the model loader.

When the model is loaded, the database entry ("person") was created in MySQL Workbench and the software was connected to it with the SQL Database loading the database with the 5726 entries/persons (Figure 9).

```

try:
    conn = mysql.connector.connect(host = "localhost",
                                   database = "persons",
                                   user = "test",
                                   password = "test")

    if conn.is_connected():
        print("Connected to mysql")

        addresses = open("addresses", "r").read().split("\n")
        locations = open("locations", "r").read().split("\n")
        jobs_file = open("jobs", "r").read().split("\n")
        jobs = list()
        for j in jobs_file:
            jobs.append(j.split(":")[0])

        persons = pickle.load(open("faces.p", "rb"))

        mycursor = conn.cursor()

        for person in list(persons):
            names = person.split(" ")

```

Figure 9. Partial Code of loading SQL Database File.

In the next step, the file `addFaceToTheDatabase` was created by the person/user to load his face into the database (1) and for authentication purpose (2). The coordinates of the newly entered face were stored in an array and loaded into the database of faces (Figure 10).

```

1  import pickle
2  import os
3  import face_recognition as fr
4
5  RESOURCE_PATH = 'Resources/faceToAdd/'
6
7  name = [x[2] for x in os.walk(RESOURCE_PATH)][0][0]
8
9  person_image = fr.load_image_file(RESOURCE_PATH + name)
10 person_encoding = fr.face_encodings(person_image)[0]
11
12 print (name)
13
14 persons = pickle.load(open("faces.p", "rb"))
15
16 persons[name.split('.')[0]] = person_encoding
17
18 pickle.dump(persons, open('faces.p', 'wb'))
19
20 print (persons)

```

Figure 10. Partial Code of the added face to the database file.

When all the above prerequisites are set, the developer can commence building the FR file, which is responsible for taking pictures (1), recognizing the face (2) while cropping out the face and removing everything else, analyzing the given face (3), and transforming the face into eigenvalues and coordinates (4). Later, while algorithm has all these actions fulfilled, it queries the database for matching arrays of eigenvalues with a small confidence level of 0.3. While most modern FR systems use a higher confidence level (0.53 for, e.g.), this proves that the system built by is more precise and has a smaller chance of authorizing the wrong person. After it tries to match the faces, it will send a message to the login file. This

file has a class named FR and uses four specialized functions: recognize_feed, hard_nms, approximate and searchInDatabase. Few samples of code can be seen in Figure 11.

```

def recognize_feed(self, video_capture):
    onnx_model = onnx.load(self.MODEL_PATH)
    onnx_rt_session = onnxruntime.InferenceSession(self.MODEL_PATH)
    input_name = onnx_rt_session.get_inputs()[0].name

    ret, frame = video_capture.read()

    cv2.imwrite("sample1.jpg", frame)

    if frame is not None and frame.any():
        height, width, _ = frame.shape

        img = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        img = cv2.resize(img, (640, 480))
        img_mean = np.array([127, 127, 127])
        img = (img - img_mean) / 128
        img = np.transpose(img, [2, 0, 1])
        img = np.expand_dims(img, axis=0)
        img = img.astype(np.float32)

        confidences, boxes = onnx_rt_session.run(None, {input_name: img})
        boxes = self.approximate(width, height, confidences[0], boxes[0], 0.7)

        box = boxes[0]

        x1, y1, x2, y2 = box
        frame[0:(y2 - y1), 0:(x2 - x1)] = frame[y1:y2, x1:x2]
        cv2.imwrite("sample.jpg", frame[y1-25:y2+25, x1-25:x2+25])

        match = self.searchInDatabase("./sample.jpg")

def hard_nms(self, box_probs):
    overlap_thresh = 0.3

    if len(box_probs) == 0:
        return list()

    picked = list()

    x1 = box_probs[:, 0]
    x2 = box_probs[:, 1]
    y1 = box_probs[:, 2]
    y2 = box_probs[:, 3]

    area = (x2 - x1) * (y2 - y1)
    indexes = np.argsort(y2)
    while len(indexes) > 0:
        last_index = indexes[-1]
        picked.append(last_index)
        last_index_index = len(indexes) - 1
        ignore = [last_index_index]

        for pos in range(0, last_index_index):
            current_index = indexes[pos]

            max_x1 = max(x1[last_index], x1[current_index])
            max_y1 = max(y1[last_index], y1[current_index])
            min_x2 = min(x2[last_index], x2[current_index])
            min_y2 = min(y2[last_index], y2[current_index])

            width = max(0, min_x2 - max_x1)
            height = max(0, min_y2 - max_y1)

            overlap = float(width * height) / area[current_index]

def approximate(self, width, height, confidences, boxes, probability_th):
    chosen_boxes_prob = list()

    for index in range(1, confidences.shape[1]):
        probabilities = confidences[:, index]
        mask = probabilities > probability_th
        probabilities = probabilities[mask]

        if probabilities.shape[0] != 0:
            mini_boxes = boxes[mask, :]
            box_prob = np.concatenate([mini_boxes, probabilities.reshape(-1, 1)], axis=1)
            chosen_boxes_prob.append(self.hard_nms(box_prob))

    if chosen_boxes_prob:
        chosen_boxes_prob = np.concatenate(chosen_boxes_prob)
        chosen_boxes_prob[:, 0] *= width
        chosen_boxes_prob[:, 1] *= height
        chosen_boxes_prob[:, 2] *= width
        chosen_boxes_prob[:, 3] *= height
        return chosen_boxes_prob[:, :4].astype(np.int32)

    return np.array([])

```

Figure 11. Code Samples of FR class.

Lastly, the message from the FR class is sent to the login class where it will be validated if the face matches a database record (1) and if the name matches the face (2). When both conditions are met, the app will authorize the user and print a welcome message. Else, when the validation process fails the app will unauthorize the user attempt to log in and terminate the process. The login class example can be seen in Figure 12.

This study provides new insights into FR code, as it uses a k means class (Figure 12) that investigates the facial color of the person to generate the skin color. This information facilitates faster search, thus optimizes the database queries. Future developments of the current application will bring contributions towards age, race, and gender evaluation.

When the application is used for analyzing the skin color, improves the search process within the database. Thus, this function improves the speed of matching the right record because it searches among the entries from the same skin color making the search by 50% faster. This mechanism also increases the security of the application. A code sample of this class can be seen in Figure 13.

```

import cv2
import facial_recognition as fr

recognizer = fr.facial_recognition()

def login():
    resultedName = recognizer.recognize_feed(video_capture= cv2.VideoCapture(0))

    return validate(resultedName)

def validate():
    resultedName = recognizer.recognize_feed(video_capture= cv2.VideoCapture(0))
    print(resultedName)
    if resultedName == "George Husac":
        return (True, resultedName)
    return (False, resultedName)

valid = validate()

if valid[0] == True:
    print('Welcome to the App ' + valid[1] + '!')
else: print(["Unauthorized Access"])

```

Figure 12. Login class.

```

def k_means(img_path):
    k = 1

    img = Image.open(img_path, 'r')

    width, height = img.size
    basewidth = 300

    if width > basewidth:
        wpercent = (basewidth/float(img.size[0]))
        hsize = int((float(img.size[1])*float(wpercent)))
        height = hsize
        width = basewidth
        img = img.resize((basewidth, hsize), Image.ANTIALIAS)

    pixels_value = list(img.getdata())

    miu = k_means_pp(pixels_value, k)
    former_miu = [None] * k
    clusters = list()
    former_clusters = list()
    for i in range(k):
        clusters.append(list())
    J = 0
    previous_J = (None, None, None)

    iter = 0
    shouldContinue = True

    while shouldContinue:

```

Figure 13. K Means Class.

4. Discussion

The present research extracts the essential information (technologies, algorithms, programming environments, functions, etc.) from the specialized literature towards the development of a FR application. Consequently, the central thesis of this paper is the development of a customized FR application, with skin color features for increasing the search speed, and its application in various fields including financial sector. Therefore, in the following Sections 4.1–4.4 several potential implementations of the developed application will be illustrated and delivered as case studies that can be used in classes in the teaching process of technical disciplines.

4.1. Authorization Access for Employees, in Any Domain

One potential real-world application of this study is the replacement of RFID access cards used in offices around the world to authorize employees access in the office buildings. In this case, the system can be implemented at the entrance doors and the software will be a cost-efficient solution and a safer alternative.

According to a study [57] in the RFID Journal, the market for RFID cards, readers, and software is expected to reach 10.7 billion USD by the end of 2022 and 17.4 billion USD by 2026. These predicaments are announced by similar websites in the RFID industry, stating that this sector is continuously growing. Our implementation can serve companies, hospitals, airports, stores, shopping malls, educational institutions and basically any entity that need to restrict access of people to rooms or buildings.

A fast cost calculation shows that the median price as revealed by the RFID Journal [57] of a low-frequency reader type and a circuit board, can cost around 100 USD, while a fully complete standalone reader can cost up to 750 USD. The active tags are 15–25 USD depending on the packaging of the tag and if it is labeled. Taking into consideration that the average-size company has around 5000 employees only the tag cost led to 100.000 USD without adding the cost of the receivers. In addition, the maintenance of the whole RFID system can become expensive, and the companies would need a verified company to do this. In the meantime, a camera to be attached to the door is around 25 USD and this cost is not even necessary for companies as most of them already have implemented some sort of video surveillance.

Surveillance cameras can be used as the technology that captures the image of the person that require institutional access, and the only cost remaining would be that of a PC (laptop, desktop computer) and a database connected to it.

Such a solution can save companies billions of USD and increase security check as the RFID tags have a major flaw, anyone with the tag can enter and the system cannot verify if it is truly the person that is authorized to enter the facility. A conceptual implementation scheme is included in Figure 14.

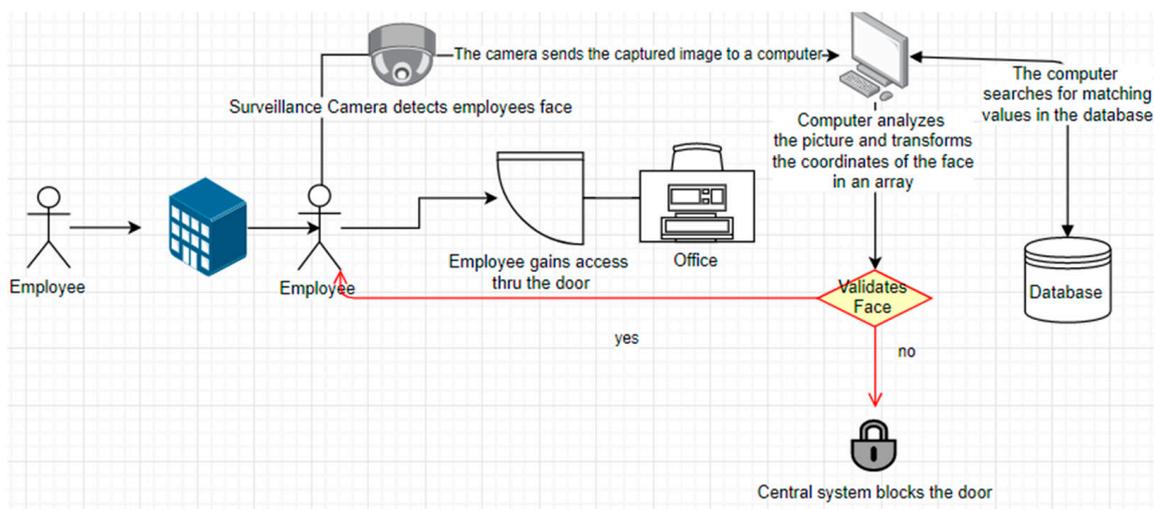


Figure 14. The author's vision on the implementation of an authorization system for employees.

4.2. Airport Security

Customs and border screening is one of the most unpleasant aspects of any traveler's trip. Passenger embarkation is one of the numerous obstacles to air transportation security checks. Indeed, such a proposal cannot fully replace the need for physical presence of customs agents because the FR software is flawless in 99% of the cases but such implementation will speed up the authentication process. In such an implementation, the customs agents will deal only with the software errors (around 1% of the cases) while the vast

majority will pass. The literature states that human error is much higher than software error. Thus, the proposed implementation made possible throughout our application will make airports a lot more efficient in authenticating the passengers. Briefly, a solution based on our app will require the following steps: user scan the passport (1), and then step in front of a camera (2) to pass the photo verification and name match (3). If the last step is successful, the passenger is allowed to pass the security area. Contrary, will returned and be processed by a human customs employee. As a result, the time required by the identification process will be reduced substantially.

4.3. Law Enforcement

FR software can help law enforcement agencies to identify and, later, catch fugitive in a lot of different situations. For example, if the police are seeking for a person with a criminal record, the surveillance cameras installed in almost any city on the planet can identify the person by the FR software built in this study. Moreover, cameras installed at the country borders can prevent a fugitive from exiting a country if they are banned. Another use case for this system related to law enforcement is that it can help track phone thieves. For example, if a user's phone is stolen, the software can be set to take a picture of the one who stole the phone. When the thief tries to use the phone, his photo is uploaded automatically to the phone's cloud. Having the photo of the prospective thief, the phone owner can ask authorities for help and identify the thief.

4.4. Financial Sector

Regarding the financial sector, the users of financial or other types of customer-oriented applications are willingly giving their fingerprints, facial scans, or other biometric information. This proves that in the commercial sector of these technologies the advancement is purely oriented around security and effectiveness. It is visible from the market trend that increased applications are implementing FR software as a security measure. Commercial banks are the most common financial institutions. Banking used to be carried out in huge rooms of buildings, which was a time-consuming chore for both clients and banking staff, but today people choose to use online banking instead of going to the bank, which makes it easier for users to complete transactions. The pandemic context and the growing popularity and demand of online banking and mobile applications forced commercial banks to invest in technology and IT development. Most of these financial institutions developed applications to make financial operations easier and more efficient in matters of time and costs. However, due to a significant increase in fraudulent operations, there has been a perceived lack of security in the network as demand for online banking has grown. According to the numbers published by the FBI and backed by Cybersecurity Ventures and Accenture the losses of victims of internet crime are estimated at 4.2 billion dollars. In addition, The Federal Communications Commission reported that 40% of theft in 2020 in the U.S.A. involved smartphones, this can be reduced by using FR software and can help law enforcement rapidly identify the thieves.

One of the many ways that financial institutions may improve security and accessibility is through FR. Most of them are using, now, this biometric authentication method which proves that the technology is useful and secure. This scientific study proposes to examine the use of FR as a method of authentication across the financial sector of this technology with the focus on tracking unauthorized access in a financial application. The suggested system's major goal is to provide secure banking authentication to end users while also tracking fraudulent login operations. This will be achieved by understanding the software and the literature related to it and creating an application that can easily recognize faces to grant permission in a financial application as well as to keep track of unauthorized access in that application by sending the ungranted permissions with the captured photo to a database that can easily be accessed from another device.

5. Conclusions

The aim of the present research was to examine the literature review in the FR domain for identifying the main algorithms, programming environments and domains that require human authentication. The results of the literature analysis were fructified in the development of a customized FR solution. Several case studies where such an application can be implemented were presented in Section 4.

This article's primary concepts are simplicity and efficiency in code development, and it applies the principles of clean code presented by Robert C. Martin in [58,59]. The developed application can be used with a device capable of establishing a person's identification and it is designed to overlay extra information about the scanned individual with the user's vision. The software was built in order to achieve FR proved to be efficient, reliable, and useful for future adaptations. The accuracy rate is 96% and is among the best according with the study included in Table 1. Although other studies have found higher accuracy, they lack the real-time photo shoot and authorization that the present study provides.

In this material, the solution was explained, and the positive or negative arguments were illustrated. Furthermore, the fundamental algorithms of this technology were explained and detailed. The proposed system used a Single-Shot Detector model that was utilized to identify the boundary coordinates of the items within the picture for FR. As mentioned in Section 3, the proposed system uses a Visual Studio Code editor, was written in Python programming language with the necessary libraries to fulfill its purpose, used an ONNX trained ML model and a MySQL Workbench Database in order to store the information provided.

Section 4 displays numerous examples of how the proposed method may be applied in the real world while providing substantial financial advantages. The following are only a few of the contributions this paper makes to the field:

- firstly, such a solution proves to be smart, safe, and a very affordable access control tool that can be applied in any business or entity throughout the world;
- secondly, this software can facilitate the embarking process safer, quicker, and more efficient in any airport;
- thirdly, the proposed system can help law enforcement agencies to prevent criminal activities and to catch fugitive persons faster by reducing the costs and the time used by the Law Enforcement Agencies;
- fourthly, this application can prove to be very efficient in the banking system and in smart retail domains, too.

Considering the use cases of this application in the financial sector, the resources that it will save in any of the domains listed, and the fact that it will make any FR operation safer, cheaper, and more effective proves that this software has utility in the real world.

Future development of code will bring additional features, such as age, race, and gender recognition.

Author Contributions: Conceptualization, V.-D.P. and G.H.; methodology, V.-D.P.; software, G.H.; investigation, G.H.; resources, G.H.; writing—original draft, G.H.; writing—review & editing, V.-D.P.; project administration, V.-D.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pati, R.; Pujari, A.K.; Gahan, P. Face recognition using particle swarm optimization based block ICA. *Multimed. Tools Appl.* **2021**, *80*, 35685–35695. [[CrossRef](#)]
2. *Face Processing: Advanced Modeling and Methods*, 1st ed.; Zhao, W.; Chellappa, R. (Eds.) Academic Press: Cambridge, MA, USA, 2011; pp. 15–36.

3. Samaria, F.S.; Harter, A.C. Parameterisation of a stochastic model for human face identification. In Proceedings of the 1994 IEEE Workshop on Applications of Computer Vision, Sarasota, FL, USA, 5–7 December 1994; pp. 138–142.
4. AT&T Laboratories Cambridge. The Database of Faces. Available online: <https://cam-orl.co.uk/facedatabase.html> (accessed on 1 August 2022).
5. Clarivate Web of Science. Available online: <https://www.webofscience.com> (accessed on 21 July 2022).
6. Monkeylearn. No-Code Text Analytics. Available online: <https://monkeylearn.com/> (accessed on 20 July 2022).
7. Bowyer, K.W.; Chang, K.; Flynn, P. A survey of approaches and challenges in 3D and multi-modal 3D+2D face recognition. *Comput. Vis. Image Underst.* **2006**, *101*, 1–15. [[CrossRef](#)]
8. Bowyer, K.W.; Chang, K.; Flynn, P. A survey of 3D and multi-modal 3D + 2D face recognition. In Proceedings of the International Conference on Pattern Recognition (ICPR) 2004, Cambridge, UK, 26 August 2004.
9. Abate, A.F.; Nappi, M.; Riccio, D.; Sabatino, G. 2D and 3D face recognition: A survey. *Pattern Recognit. Lett.* **2007**, *28*, 1885–1906. [[CrossRef](#)]
10. Cheng, Z.; Zhu, X.; Gong, S. Face re-identification challenge: Are face recognition models good enough? *Pattern Recognit.* **2020**, *107*, 107422. [[CrossRef](#)]
11. Krishna, I.M.V.; Kanth, R.M.; Sowjanya, V. Machine Learning Based Face Recognition System. *ECS Trans.* **2022**, *107*, 19979. [[CrossRef](#)]
12. Lee, G. Fast and more accurate incremental-decremental principal component analysis algorithm for online learning of face recognition. *J. Electron. Imaging* **2021**, *30*, 043012. [[CrossRef](#)]
13. Gang, A.; Bajwa, W.U. A linearly convergent algorithm for distributed principal component analysis. *Signal Process.* **2022**, *193*, 108408. [[CrossRef](#)]
14. Kumar, R.S. Principal Component Analysis: In-Depth Understanding through Image Visualization. Available online: <https://towardsdatascience.com/principal-component-analysis-in-depth-understanding-through-image-visualization-892922f77d9f> (accessed on 19 July 2022).
15. Derksen, L.; Xifara, D. Visualising High-Dimensional Datasets Using PCA and t-SNE. Available online: <http://luckyllwk.github.io/2015/09/13/visualising-mnist-pca-tsne/> (accessed on 19 July 2022).
16. Pan, Z.; Healey, G.; Prasad, M.; Tromberg, B. Face recognition in hyperspectral images. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 1552–1560. [[CrossRef](#)]
17. Medioni, G.; Waupotitsch, R. Face modeling and recognition in 3-D. In Proceedings of the 2003 IEEE International SOI Conference, Nice, France, 17 October 2003.
18. Lu, X.; Jain, A.K.; Colbry, D. Matching 2.5D face scans to 3D models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *28*, 31–43. [[CrossRef](#)]
19. Chang, K.J.; Bowyer, K.W.; Flynn, P.J. Effects on facial expression in 3D face recognition. In *Biometric Technology for Human Identification II, Proceedings of the Defense and Security Conference, Orlando, FL, USA, 28 March–1 April 2005*; Jain, A.K., Ratha, N.K., Eds.; Society of Photo-Optical Instrumentation Engineers: Bellingham, WA, USA, 2005; Volume 5779, pp. 132–143.
20. Tsalakanidou, F.; Tzovaras, D.; Strintzis, M.G. Use of depth and color eigenfaces for face recognition. *Pattern Recognit. Lett.* **2003**, *24*, 1427–1435. [[CrossRef](#)]
21. Papatheodorou, T.; Rueckert, D. Evaluation of automatic 4D face recognition using surface and texture registration. In Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Seoul, Republic of Korea, 19 May 2004; pp. 321–326. [[CrossRef](#)]
22. Bronstein, A.M.; Bronstein, M.M.; Kimmel, R. Expression-invariant 3D face recognition. In *Audio- and Video-Based Biometric Person Authentication*; Kittler, J., Nixon, M.S., Eds.; Lecture Notes in Computer Science Volume 2688; Springer: Berlin/Heidelberg, Germany, 2003; pp. 62–70. [[CrossRef](#)]
23. Beumier, C.; Acheroy, M. Automatic 3D face authentication. *Image Vis. Comput.* **2000**, *18*, 315–321. [[CrossRef](#)]
24. Wang, Y.; Pan, G.; Wu, Z.; Han, S. Sphere-Spin-Image: A Viewpoint-Invariant Surface Representation for 3D Face Recognition. In *Computational Science–ICCS 2004, Proceedings of the 4th International Conference on Computational Science (ICCS 2004), Kraków, Poland, 6–9 June 2004*; Bubak, M., van Albada, G.D., Sloot, P.M.A., Dongarra, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; pp. 427–434.
25. Lu, R.; Zhu, F.; Hao, Y.; Wu, Q. Simple and efficient improvement of spin image for three-dimensional object recognition. *Opt. Eng.* **2016**, *55*, 113102. [[CrossRef](#)]
26. Papoulis, A.; Pillai, S.U. *Probability, Random Variables, Stochastic Processes*; McGraw Hill: New York, NY, USA, 2002; pp. 120–151.
27. Haykin, S. *Neural Networks: A Comprehensive Foundation*; Macmillan Publishing: New York, NY, USA, 1999.
28. Bera, S.; Shrivastava, V.K. Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification. *Int. J. Remote Sens.* **2020**, *41*, 2664–2683. [[CrossRef](#)]
29. Eleyan, A.; Demirel, H. PCA and LDA Based Neural Networks for Human Face Recognition. In *Face Recognition*; Delac, K., Grgic, M., Eds.; IntechOpen: London, UK, 2007; pp. 93–106. [[CrossRef](#)]
30. Delac, K.; Grgic, M.; Grgic, S. Image compression effects in face recognition systems. In *Face Recognition*; Delac, K., Grgic, M., Eds.; IntechOpen: London, UK, 2007; pp. 75–92.

31. Sharma, R.; Patterh, M.S. A Systematic Review of PCA and Its Different Form for Face Recognition. *Int. J. Sci. Eng. Res.* **2014**, *5*, 1306–1309.
32. Bazama, A.; Mansur, F.; Alsharif, N. Security System by Face Recognition. *AlQalam J. Med. Appl. Sci.* **2021**, *4*, 58–67.
33. Patel, V.M.; Chen, Y.-C.; Chellappa, R.; Phillips, P.J. Dictionaries for image and video-based face recognition. *J. Opt. Soc. Am. A* **2014**, *31*, 1090–1103. [[CrossRef](#)]
34. Sirovich, L.; Kirby, M. Low-dimensional procedure for the characterization of human faces. *J. Opt. Soc. Am. A* **1987**, *4*, 519–524. [[CrossRef](#)]
35. ORL: Our Database of Faces by AT&T Laboratories Cambridge. Available online: <https://www.v7labs.com/open-datasets/orl> (accessed on 10 June 2022).
36. Bing, H.; Xianfeng, H.; Ruizhen, H. Research of Face Detection Based on AdaBoost and ASM. *Open Cybern. Syst. J.* **2014**, *8*, 183–190. [[CrossRef](#)]
37. Gormley, M. Lecture Notes on Introduction to Machine Learning: PCA + AdaBoost. 2018. Available online: <http://www.cs.cmu.edu/~jmgormley/courses/10601-s18/slides/lecture30-pca-adaboost.pdf> (accessed on 10 June 2022).
38. Mahmood, Z.; Ali, T.; Khattak, S.; Khan, S.U. A comparative study of baseline algorithms of face recognition. In Proceedings of the 2014 12th International Conference on Frontiers of Information Technology, Islamabad, Pakistan, 17–19 December 2014; pp. 263–268.
39. Ehlers, A.; Baumann, F.; Spindler, R.; Glasmacher, B.; Rosenhahn, B. PCA enhanced training data for adaboost. In *Computer Analysis of Images and Patterns, Proceedings of the 2011 International Conference on Computer Analysis of Images and Patterns (CAIP 2011), Seville, Spain, 29–31 August 2011*; Real, P., Diaz-Pernil, D., Molina-Abril, H., Berciano, A., Kropatsch, W., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 410–419.
40. Kao, I.-H.; Chan, C.-Y. Comparison of Eye and Face Features on Drowsiness Analysis. *Sensors* **2022**, *22*, 6529. [[CrossRef](#)]
41. Yang, Y. Smart community security monitoring based on artificial intelligence and improved machine learning algorithm. *J. Intell. Fuzzy Syst.* **2020**, *38*, 7351–7363. [[CrossRef](#)]
42. He, D.; He, X.; Yuan, R.; Li, Y.; Shen, C. Lightweight network-based multi-modal feature fusion for face anti-spoofing. *Vis. Comput.* **2022**. [[CrossRef](#)]
43. Wang, C.; Xu, S.; Yang, J. Adaboost Algorithm in Artificial Intelligence for Optimizing the IRI Prediction Accuracy of Asphalt Concrete Pavement. *Sensors* **2021**, *21*, 5682. [[CrossRef](#)]
44. Natras, R.; Soja, B.; Schmidt, M. Ensemble Machine Learning of Random Forest, AdaBoost and XGBoost for Vertical Total Electron Content Forecasting. *Remote Sens.* **2022**, *14*, 3547. [[CrossRef](#)]
45. Ahmad, I.; Ul Haq, Q.E.; Imran, M.; Alassafi, M.O.; AlGhamdi, R.A. An Efficient Network Intrusion Detection and Classification System. *Mathematics* **2022**, *10*, 530. [[CrossRef](#)]
46. Ding, Y.; Zhu, H.; Chen, R.; Li, R. An Efficient AdaBoost Algorithm with the Multiple Thresholds Classification. *Appl. Sci.* **2022**, *12*, 5872. [[CrossRef](#)]
47. Crumpler, W. How Accurate are Facial Recognition Systems—and Why Does It Matter? Strategic Technologies Blog. 2020. Available online: <https://www.csis.org/blogs/technology-policy-blog/how-accurate-are-facial-recognition-systems-%E2%80%93-and-why-does-it-matter> (accessed on 8 October 2022).
48. Yin, C.; Zhu, Y.; Fei, J.; He, X. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE Access* **2017**, *5*, 21954–21961. [[CrossRef](#)]
49. Jia, Y.; Wang, M.; Wang, Y. Network intrusion detection algorithm based on deep neural network. *IET Inf. Secur.* **2019**, *13*, 48–53. [[CrossRef](#)]
50. Vinayakumar, R.; Alazab, M.; Soman, K.P.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S. Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access* **2019**, *7*, 41525–41550. [[CrossRef](#)]
51. Kasongo, S.M.; Sun, Y. A Deep Learning Method With Filter Based Feature Engineering for Wireless Intrusion Detection System. *IEEE Access* **2019**, *7*, 38597–38607. [[CrossRef](#)]
52. Kanimozhi, V.; Jacob, P. UNSW-NB15 dataset feature selection and network intrusion detection using deep learning. *Int. J. Recent Technol. Eng.* **2019**, *7*, 443–446.
53. Mahalakshmi, G.; Uma, E.; Aroosiya, M.; Vinitha, M. Intrusion Detection System Using Convolutional Neural Network on UNSW NB15 Dataset. In *Advances in Parallel Computing Technologies and Applications*; Hemanth, D.J., Elhosney, M., Nguyen, T.N., Lakshmann, S., Eds.; IOS Press: Amsterdam, The Netherlands, 2021; Volume 40, p. 1.
54. Fu, Y.; Du, Y.; Cao, Z.; Li, Q.; Xiang, W. A Deep Learning Model for Network Intrusion Detection with Imbalanced Data. *Electronics* **2022**, *11*, 898. [[CrossRef](#)]
55. Mijalkovic, J.; Spognardi, A. Reducing the False Negative Rate in Deep Learning Based Network Intrusion Detection Systems. *Algorithms* **2022**, *15*, 258. [[CrossRef](#)]
56. Lin, C.-L.; Huang, Y.-H. The Application of Adaptive Tolerance and Serialized Facial Feature Extraction to Automatic Attendance Systems. *Electronics* **2022**, *11*, 2278. [[CrossRef](#)]
57. Gough, S. Current RFID Trends and Challenges You Should Know About, in RFID JOURNAL LIVE! 2021. Available online: <https://rfidjournallive.com/content/blog/current-rfid-trends-and-challenges-you-should-know-about/> (accessed on 22 August 2022).

-
58. Martin, R.C. *Clean Code: A Handbook of Agile Software Craftsmanship*; Pearson Education: London, UK, 2009.
 59. Martin, R.C. *Clean Code-Refactoring, Patterns, Testen und Techniken für sauberen Code: Deutsche Ausgabe*; MITP-Verlags GmbH & Co. KG: Bonn, Germany, 2013. (In German)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.