A security system with automated person identification based on computer vision technologies*

Bohdan Zhurakovskyi[1,*,†], Viktor Shutenko[1,†], Anatoliy Makarenko[1,†], Oleksandr Pliushch[2,†], and Rostyslav Zakharov[3,†]

[1] National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute," 37 Peremogy ave., 03056 Kyiv, Ukraine
[2] Taras Shevchenko National University of Kyiv, 60 Volodymyrska str., 01601 Kyiv, Ukraine
[3] State University of Trade and Economics, 19 Kyoto str., 02156 Kyiv, Ukraine

Abstract

This article is dedicated to the development of a software system that performs person identification based on facial images using modern computer vision methods while addressing requirements for accuracy, speed, and adaptability. The following research is conducted in this work: analysis of contemporary biometric identification systems to determine their advantages and limitations; investigation of computer vision algorithms applied for face recognition and the justification of their selection; definition of system functional requirements; design of the software solution architecture and the database structure; implementation of the system's key modules according to the defined requirements; system testing and performance evaluation. A fully functional software complex has been created, encompassing everything from the video capture module to person detection and recognition. A crucial feature is the integration of an anti-spoofing module with two verification levels—neural network-based and behavioral—which allows for reliable detection of fraudulent attempts. The developed software is suitable for practical implementation in controlled local conditions, namely at enterprises that require automated access control in real time. Moreover, thanks to its autonomy and flexible architecture, the system can be deployed even on robotic platforms for real-time access monitoring. This research also opens up prospects for further development in the areas of additional biometrics and improving verification algorithms.

Keywords

computer vision algorithms, biometric identification, recognition process, image analysis, security software, neural network, anti-spoofing, caching, vector indexing caching, vector indexing

1. Introduction

As the days pass, we feel the increasing importance of security more acutely, both on a physical level and in the digital space. Intelligent access control systems are becoming particularly relevant. They offer a high-quality alternative to simply recognizing people by a password or conventional keys; instead, they use biometric data, which makes the system even more reliable, increases user confidence, and significantly simplifies life through the automated management of access to various facilities and data.

Thus, one of the most promising areas in this field is the use of computer vision for face identification. This approach ensures high speed, efficiency, and convenience in the identity verification process, and can be integrated into various systems—from corporate offices and schools to industrial sites. Although this technology is developing quite actively, certain obstacles remain on the path to its widespread implementation and use. Among these challenges are ensuring operational accuracy in various conditions, increasing the system's security level against unauthorized access attempts, and the ability to adapt and update user data. That is why the

* CPITS-II 2025: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, October 26, 2025, Kyiv, Ukraine
* Corresponding author.
† These authors contributed equally.
zhurakovskybiyu@tk.kpi.ua (B. Zhurakovskyi); victor.shutenko@gmail.com (V. Shutenko); makarenkoa@ukr.net (A. Makarenko); opliusch@yahoo.com (O. Pliushch); zrost19@gmai.com (R. Zakharov)
0000-0003-3990-5205 (B. Zhurakovskyi); 0009-0002-3537-4101 (V. Shutenko); 0000-0002-4081-328X (A. Makarenko); 0000-0001-5310-0660 (O. Pliushch); 0009-0003-1433-7755 (R. Zakharov)

development of a system that not only recognizes a person but also verifies their uniqueness and authenticity is a relevant and practically significant task.

The research task is to create a software system capable of identifying individuals by facial image with a given accuracy and stability of operation. The system must process the input image, extract a face from it, compare it with existing standards in the database and return the identification result. In addition, it must ensure the storage of results in the database, allow management of the composition of standards and interact with the user through the interface. Thus, the creation of an effective personal identification system based on facial image analysis is a complex interdisciplinary task that requires the involvement of modern computer vision methods, data processing algorithms and software design principles. A comprehensive solution to this problem will contribute to increasing the reliability of automated security systems and opens up prospects for further research in the field of biometric authentication [1].

Thus, the creation of an effective facial recognition system based on facial image analysis is a complex interdisciplinary task that requires the involvement of modern computer vision methods, data processing algorithms, and software design principles. A comprehensive solution to this problem will contribute to increasing the reliability of automated security systems and opens up prospects for further research in the field of biometric authentication.

2. Description of the subject area

2.1. Using biometrics for access control

Access control is perhaps the most important element of creating secure conditions in information and physical systems. As the speed of technological development is rapidly increasing, the need for more reliable and convenient ways to identify individuals who have the right to access certain resources is also increasing. Today, we know such traditional methods as: passwords, physical keys and their modifications, PIN codes. However, they have quite significant drawbacks—they can be forgotten, lost, transferred or forged. Given this, the optimal solution for ensuring confidentiality has become biometric systems based on the unique characteristics of a person. Biometric data are unique measurable physiological or behavioral characteristics of a person that can be used for automated face recognition [2]. Thus, face identification occurs by comparing the input data with previously entered data in the database.

Biometric systems, unlike other well-known authentication methods, directly use the characteristics of the person, which cannot be easily lost, forgotten or transferred to others [3], so the user is provided with a higher level of security and comfort in use.

The modern market is rich in various identification methods, for the implementation of which developers use their own set of technological tools. Let's consider the main ones in more detail, taking into account their properties:

- By fingerprints

This method is used in smartphones, ATMs, corporate systems (for example, scanners in Dell offices). In this case, optical or capacitive scanners are used that read the unique pattern of papillary lines on the skin of the fingers. Based on this, certain digital templates are formed. This method is famous for its speed of information processing and is affordable in terms of the cost of equipment (scanners), but the presence of damage or other factors that affect the display of a fingerprint can reduce efficiency.

- By the iris

Used in border control (for example, in the UAE) and highly secure facilities. Identification of users by the iris is possible due to its complex and unique structure, which is formed prenatally and remains unchanged throughout life. Scanning the eye occurs using the infrared range, which

allows you to create a detailed template of it. The disadvantages of this method include the need for special equipment, which causes difficulties in integration into real systems, although it is distinguished by high accuracy.

- By palm geometry or vein pattern

Already used in medical institutions in Japan to access confidential data. These identification methods using specialized equipment can recognize a person by analyzing the physical parameters of the hand, its three-dimensional structure or special vein patterns under the skin. Unlike fingerprint identification, this method is more resistant to external damage and has an acceptable level of accuracy, but scaling to a large number of people may have certain limitations.

- Voice

A behavioral feature used in banking call centers (e.g. Barclays Voice ID). This identification technology uses digital processing to analyze the acoustic features of the voice: intonation, frequency, timbre. Work can be affected by changes in a person's voice due to health conditions or other physical conditions. The method is also sensitive to external noise, which negatively affects the quality of work.

- By face

Used in airports, smartphones (Face ID technology in iPhone), video surveillance systems. Systems based on this method of user recognition work by processing visual information from images or videos. This allows you to create a vector description of the main features of a person's face. Modern computer vision and deep learning algorithms provide high accuracy even under typically unfavorable conditions, such as poor lighting, partial face coverage, and head movements [4].

Among the listed methods, one of the most popular and in practice the most convenient to use is the facial recognition method, since it does not require direct contact, processes information quickly, and the cameras necessary for its implementation are already equipped in most environments [5].

Biometric identification systems for access control can be implemented both in physical objects (for example, enterprises, educational institutions or airports) and in digital environments, such as online banking, corporate platforms or mobile applications. In particular, facial recognition technology, which is widely used by modern manufacturers to unlock devices or applications, is an example of its effectiveness and high level of user trust [6].

Biometric systems have a number of advantages, including: high uniqueness of features, difficulty of their forgery and automated identification process. Due to this, they are mainly chosen for use in the security sector [7]. At the same time, there are certain difficulties, such as the need to protect the confidentiality of biometric data, sensitivity to shooting conditions (lighting, viewing angle) and ensuring protection against spoofing, i.e. imitation of someone else's face using images or videos [8].

The use of biometric data for access control is one of the most promising areas in the development of modern security systems. Despite the technical and ethical challenges, improving algorithms, introducing new standards and integrating with IoT create conditions for the implementation of continuous and adaptive authentication systems that can meet the requirements of the future [9].

To implement the system, facial recognition was chosen as the main method of identifying individuals, since this approach does not require physical contact and is easily integrated into existing infrastructure, such as webcams and surveillance cameras. In addition, it provides high data processing speed and achieves a harmonious combination of high security and ease of use.

2.2. Architecture and principles of operation of facial recognition systems

The face recognition system belongs to the category of biometric systems that use the analysis of a person's appearance and characteristics for identification or verification. From a technical perspective, these systems utilize computer vision and machine learning algorithms.

The operating principle is as follows: first, they automatically locate the face in the image, then convert it into a numerical vector that describes its unique characteristics, and finally, compare this vector with previously stored data in the database.

As previously mentioned, the main advantage of this method is the absence of physical contact and the possibility of its implementation using ordinary surveillance cameras, which are already installed at the vast majority of facilities that might require user identification.

Modern facial recognition systems are based on a complex multi-layer architecture that combines hardware and software algorithms for effective face identification. This allows processing large amounts of data in real time, ensuring high accuracy and scalability [10]. I would also like to draw attention to their versatility—the systems work reliably even in poor lighting conditions, at different angles and with partial overlap. The main components of such systems are:

Image capture module—responsible for receiving video or images from the camera in real time. At this stage, it is extremely important to obtain high-quality input data, as this directly affects the accuracy of recognition.

Face detection module—finds faces in the image among other objects. Methods such as Haar Cascade Classifier or neural networks such as MTCNN (Multi-task Cascaded Convolutional Neural Network) [11] are often used here. The detected parts are cropped and passed to the next module for further processing.

Normalization module—corrects the position, lighting, and scale of the image to prepare the face for processing. This stage allows you to reduce the impact of external factors on the quality of recognition.

Feature extraction module—builds a vector of facial features. For example, in the FaceNet model, the face is converted into a 128-dimensional vector [12]. This vector is a unique projection of the face, which allows you to accurately distinguish one person from another.

Matching module—calculates the distance between the vectors of the new image and those recorded in the database. If the distance is less than a certain threshold, the person is considered identified. In practice, Euclidean or cosine distance is usually used to calculate similarity.

Decision-making module—based on the comparison result, the system makes a decision to grant or deny access.

Systems can be implemented both in a local environment and on the basis of cloud platforms—the choice depends on specific tasks, goals, security level and available resources. Local solutions are usually chosen in cases where full control over the collected data is critically important [13]. This is especially true in conditions of strict confidentiality. However, local environments require serious investments in their own infrastructure, regular software updates and a large amount of powerful equipment, which is not always justified or possible. Cloud systems, on the contrary, attract with their flexibility, ease of scaling and availability. Thanks to the use of a cloud environment, it is easy to integrate the system into other services, conduct real-time monitoring or work simultaneously on different devices regardless of their hardware characteristics. At the same time, cloud solutions require a stable connection to the Internet, and data storage on external servers is accompanied by potential risks of leakage. Both approaches have their advantages and disadvantages, so in practice, hybrid architectures are often found, where local modules work together with cloud ones to achieve a balance between security and performance.

Facial recognition systems for access control are usually supplemented with auxiliary modules. These can be event logging systems that allow for activity auditing; an administrator interface for responding to unauthorized login attempts, and notification modules for responding to situations with suspicious activity. Some systems have an access filtering function, which is especially

important for large enterprises with a branched structure, as this allows them to be adapted to security tasks [14, 15].

Therefore, the architecture of such systems should be designed as flexible, modular solutions with the ability to scale. They should not only accurately identify the user, but also integrate computer vision algorithms with decision-making and data storage mechanisms to ensure accuracy and speed.

2.3. Review and analysis of existing analogues

The current market for facial recognition systems offers a wide range of commercial solutions and open source projects that differ significantly in their functionality, level of accuracy, resource consumption, and areas of application. This section provides a comparative analysis of the most common systems, which will allow us to determine their advantages, disadvantages, and optimal areas of use.

First, let's look at the most famous commercial systems:

Apple Face ID

This technology from Apple is one of the most famous implementations of facial recognition in the commercial sector. It uses deep analysis technologies (TrueDepth camera) and 3D face modeling. Its great advantage is that all data processing is performed directly on the device, which minimizes the risks of leakage of confidential information, and also guarantees high resistance to spoofing. However, this technology is available only on a limited number of Apple devices and does not support integration into external systems.

Amazon Rekognition

This cloud-based facial recognition system from AWS offers APIs for integration with third-party systems, which is a definite advantage. It also has a fairly wide functionality, which includes facial recognition, emotion analysis, finding similar faces and comparing them with databases, and supports integration with other AWS services, ensuring easy scalability. However, its operation depends on a constant Internet connection, and the use of the technology raises discussions about privacy and ethical aspects [16].

Clearview AI

A controversial technology, the implementation of which is based on a large database consisting of images illegally collected from social networks and other publicly available sources. According to reliable tests, it demonstrates high speed of work with large volumes of data, which, together with accuracy, makes it one of the favorites in the field of facial recognition. However, Clearview AI also has a significant disadvantage, which most experts note—violation of privacy through illegal accumulation of data and high risks of uncontrolled use.

At the same level as commercial systems, there are also open-source analogues that allow you to independently create custom systems without relying on third-party services:

OpenCV (cv2.face module)

The open-source OpenCV library includes the cv2.face module, which offers basic algorithms such as: Eigenfaces, Fisherfaces, and LBPH. These solutions are suitable for local projects with low accuracy requirements, as they are easy to use and do not require significant computing resources. However, compared to modern neural networks, the accuracy of such methods is limited.

Dlib + ResNet

This technology uses a 128-dimensional vector representation of faces for recognition. Dlib in combination with ResNet provides powerful recognition with high accuracy, if the data is properly prepared. The system performs well in situations where high accuracy is required, but requires significant efforts for high-quality training and computing power for optimal operation.

FaceNet / DeepFace

These open-source deep neural models—FaceNet and DeepFace, demonstrate excellent results in tests—accuracy of over 99% on standard datasets. Such a connection allows you to build quite complex recognition systems without the need for cloud environments. They are actively used in

scientific research and commercial products. And although their main advantages are high accuracy and scalability, their implementation requires significant computing resources, which can complicate integration into simple local systems.

After conducting research and comparing all the analyzed models, we can conclude that it is open-source solutions that make it possible to develop an independent and flexible system with a level of accuracy sufficient for practical application.

3. System design

3.1. General system architecture

In general, the ultimate technical goal of this work is to obtain a software solution for a local facial recognition system, which in turn creates security conditions. This system is built in a monolithic architectural style. This means that the entire application is created as a single technological system. All components are located on a single device, without interacting using external APIs or cloud environments. This implementation decision was made to increase the security and autonomy of the system.

It should also be noted that the system is based on a modular approach. The system architecture involves dividing it into logical components: capturing a video stream from a camera; detecting and highlighting faces in the frame; comparing embeddings with reference ones in the database; maintaining an event log (logging); administering users and access rights.

And although physically all these components are implemented as a single software, logical division allows us to provide the modularity we are talking about. That is, all the mentioned components have their own tasks, but they are all integrated into a single local program and work harmoniously. In turn, this is done so that in the future there is an opportunity to scale the system, expand its functionality without any cardinal changes and global restructuring of the architecture. That is, we can conclude that the system is built mainly in a procedural style with a clear division into functions (it uses a modular approach, when the main tasks are allocated into functional blocks) [17].

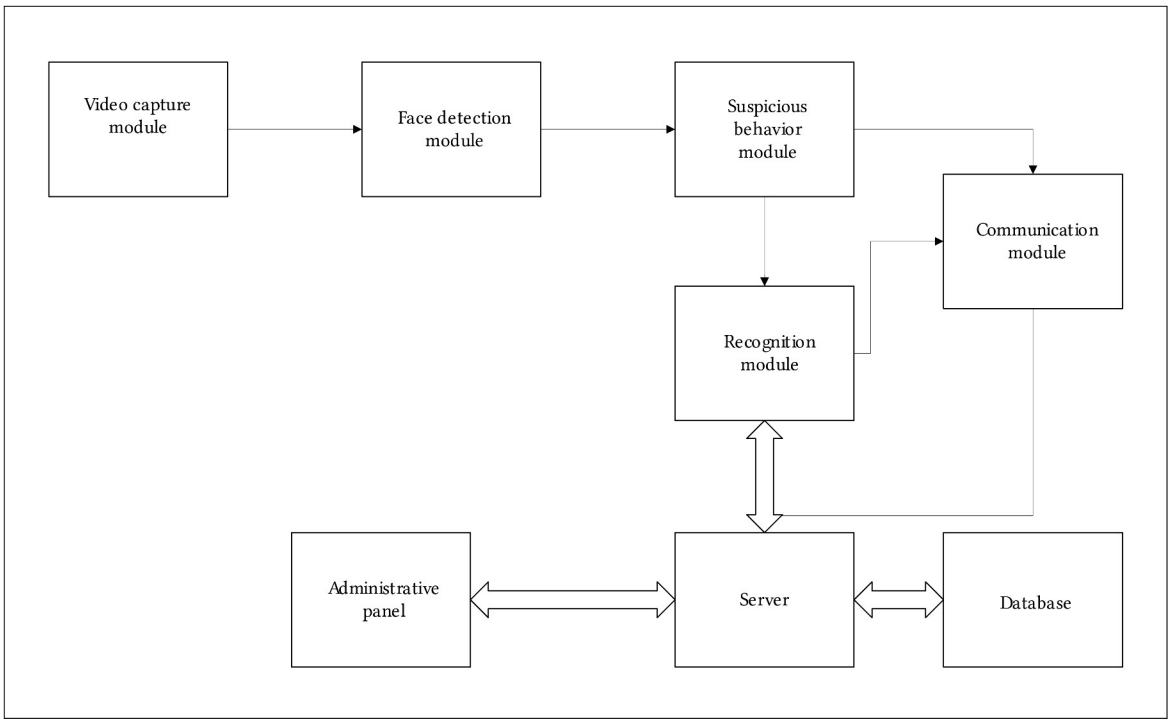

Figure 1: Structural diagram of the system

One of the following principles when creating an architecture is the possibility of real-time processing (real-time processing). Since the system has to work with a constant video stream from the camera, delays at each stage must be minimal so as not to become fatal. Also, all selected algorithms and libraries must work quickly and be optimized for fast work.

No less important is the ability to scale. Although the system is local, the possibility of expansion still remains: for example, you can add more users, increase the size of the database or connect additional modules (emotion analysis, voice recognition or other biometric data), without completely rebuilding the system core [18].

In order to prevent data leakage, the system is designed so that biometric features, namely embeddings, which carry information about unique features, are stored exclusively in a local database with strict access control [19]. Also, to ensure transparency and the possibility of security auditing, detailed logging of user actions and system events, i.e. logging [20], is provided.

Thus, taking into account all these factors, we obtain a system whose architecture is capable of providing high data processing speed [21], minimal delays and significant accuracy during recognition, which greatly simplifies the process of passing the check in a conventional office that requires serious security measures, or other institutions where this system can be integrated [22].

3.2. Module design

The development of modern facial recognition systems is a complex engineering task, therefore it requires the integration of various technologies and algorithms, which ensures high accuracy, speed and reliability. The correct organization of the components of such a system is the key to its successful functioning and further improvement. To achieve this result, it was decided to apply a modular approach to the architecture of the facial recognition system.

To ensure high flexibility, scalability and ease of system support, we will divide the system into modules, each of which is responsible for specific system functions. Their interaction creates the integrity and coherence of the entire mechanism.

This approach allows you to isolate the logic of different subsystems. This simplifies testing, makes it easier to detect and fix errors, and also provides the ability to expand functionality without a radical restructuring of the architecture.

3.2.1. Description of the module structure

To implement the functionality of a security system with automatic recognition of people, the following main modules were selected: video capture module; face processing module; face recognition module; database work and logging; anti-spoofing; administration.

It is thanks to them that we can provide full security guarantee functionality, which includes face authentication itself, administration of this process and interaction, which directly allows for dynamic communication with the system [23].

The video capture module is considered the first point of contact with the outside world. It is responsible for stable and continuous receipt of a video stream in real time from a connected camera device. In addition to basic frame acquisition, the module can also include pre-processing to improve the quality of the input signal. It is able to adjust the exposure parameters, white balance or apply simple noise reduction filters. To ensure compatibility with different types of cameras (local, network, USB), the OpenCV library is used, namely its Python wrapper cv2, which provides a universal access interface [24]. The module initializes the connection to the device and reads each video frame in BGR format, ensuring stable data transfer for further processing with minimal delays. Additionally, it is possible to set camera parameters, such as resolution, frame rate. The module is designed so that due to isolation it will be easy to integrate support for new video sources in the future, perhaps even streams from network protocols or files.

After that, we install the face processing module. Its task is to detect the presence of faces in the received frames and prepare information for further recognition. To perform this task, the face recognition library is used. It effectively uses the HOG algorithm implemented in the dlib library

[25]. HOG analyzes the distribution of pixel intensity gradients to detect characteristic facial features. This provides high speed and resistance to certain changes in lighting and head tilt angle.

Next, the found faces are determined using coordinates (bounding boxes), which allows them to be accurately localized in the frame. After that, using the OpenCV library (cv2), each found face is cropped according to the specified coordinates and scaled to the standard size required for further processing. The main actions that occur in this module also include converting the color format from BGR to RGB, which is necessary for face_recognition [26].

Another important function is face alignment based on keypoint detection using the same face_recognition. This allows you to reduce the impact of pose variations (tilts, turns) and significantly increase the accuracy of recognition. Then only the final stage is the formation of the famous embedding, which describes unique facial features and is the basis for further recognition.

The next module is the face recognition module. It is the "brain" of the system for recognizing people. Here, the numerical face print (embedding) is compared with those that the system already has. For this comparison, special tools are used that can determine how similar two values are. Thus, the fact of coincidence between the embeddings is established and the proximity distance between the vectors is calculated, which allows you to increase the accuracy and control over the recognition process [27].

After that, the module gives the desired result or reports an unknown person. If there are several faces in the frame, the system tries to focus on the main one (for example, the one in the middle or the largest) [28].

The database module is used to manage the system data. Its work includes storing users' biometric data (face embeddings), their personal information, and event logs.

Interaction with various types of relational databases is implemented through the PyODBC library, which provides a universal interface via the ODBC standard [29]. Face embeddings (complex numeric vectors) are serialized using the pickle library for convenient storage as binary data. This module supports standard data management operations. These include adding new users, searching and comparing embeddings, and maintaining event logs that capture all key actions in the system, with the mandatory use of parameterized queries to ensure data security [30].

The mandatory use of parameterized SQL queries is a key element for protecting against SQL injections and ensuring data integrity. In the future, the module can be extended to support various backup and recovery strategies.

The level of system security is extremely important in this work. This is provided by the anti-spoofing module. Its main goal is to detect attempts to deceive the system by using fake facial images [31]. Such unauthorized actions include the use of photographs, videos, or even three-dimensional masks that imitate the faces of people from the database.

At the moment, the integration of the Mediapipe library is planned. It provides tools for analyzing key facial points and tracking barely noticeable micromovements, which is characteristic of a living face [32]. It is also possible to develop and integrate custom deep learning models. This functionality can be made based on the YOLO architecture (Ultralytics library), specially trained to classify input images into "real faces" and various types of "fake" [33].

Thus, combining a variety of anti-spoofing methods within this module will allow you to create a multi-level system of protection against unauthorized access.

The last is the administration module. It provides the administrative institution with a convenient and understandable interface for performing key tasks of system management. This is managing user accounts, and viewing system logs [34].

Here, the modular structure allows you to independently develop and update the administration interface without making changes to the basic logic of recognition and data processing, which is logical when building the system.

3.2.2. Module Interaction

To ensure the coordinated, holistic operation of the system as a single complex, it is necessary to outline a clear interaction between individual modules. Each of them, performing its specific local task, in any case must interact with other modules through certain standardized interfaces. This ensures the independence of the modules, but at the same time guarantees the consistency of data processing. Let us consider in more detail the process of interaction of the modules within the framework of this study.

After starting the program, the system starts with the initialization of the image capture module. It establishes a stable connection with the camera and launches a continuous stream of frames in real time. This module also easily adapts everything to the use of different types of video cameras by changing the appropriate settings or drivers.

The captured frames are transferred to the second module—face processing. Here, the received information is analyzed for the presence of faces and this information is edited. Using the appropriate algorithms, the module determines the exact coordinates of each detected face in the frame, forms bounding boxes. Then the module crops the found face, scales it, and normalizes it.

The resulting cut-out face is processed in the anti-spoofing module. There, an analysis is made as to whether a person is alive in the frame or not, and if it turns out to be real, the data is transferred back to the processing module [35]. A vector of human features is formed in it and sent for work with it in subsequent modules.

In the face recognition module, the face recognition itself is actually performed. Its data is compared with the existing ones (the process is described in more detail in the module description). That is, in parallel, we perform a query to the database using the database work module and "extract" the necessary records from there. It also enters logging information into the log events. Hence, this module serves as the main hub for both storing biometric data and logging.

Separately, we have interaction with the administration module. The results of the work performed are transferred there [36]. In addition, the module displays all the necessary information needed to manage the system in real time.

So, having analyzed how the modules of this system interact, we can draw several conclusions. The key principles underlying this interaction are sequential data processing. This means that each module receives and processes information from the previous one, ensuring a controlled flow. In addition, the minimum level of communication between them, carried out through the transfer of standardized data, increases the system's resistance to changes in the internal implementation of individual components. Therefore, thanks to these principles, the developed architecture is flexible, reliable and ready for further development.

3.3. Database description

The full functioning of a security system with automatic face recognition requires not only a direct, high-precision identification mechanism, but also a reliable infrastructure for storing and managing all important information. That is why an integral part of the architecture is the database subsystem, which ensures the preservation of information about registered users, administrative department data, as well as detailed logs of all key actions and events occurring in the system. Such a repository is the foundation for guaranteeing stability, security, auditability and further scalability of the system.

A relational database is used to store data, which acts as a central repository for biometric and auxiliary information. The main tasks of the database are: storing user face embeddings; storing information about system users (identifiers, names, additional attributes); maintaining event logs.

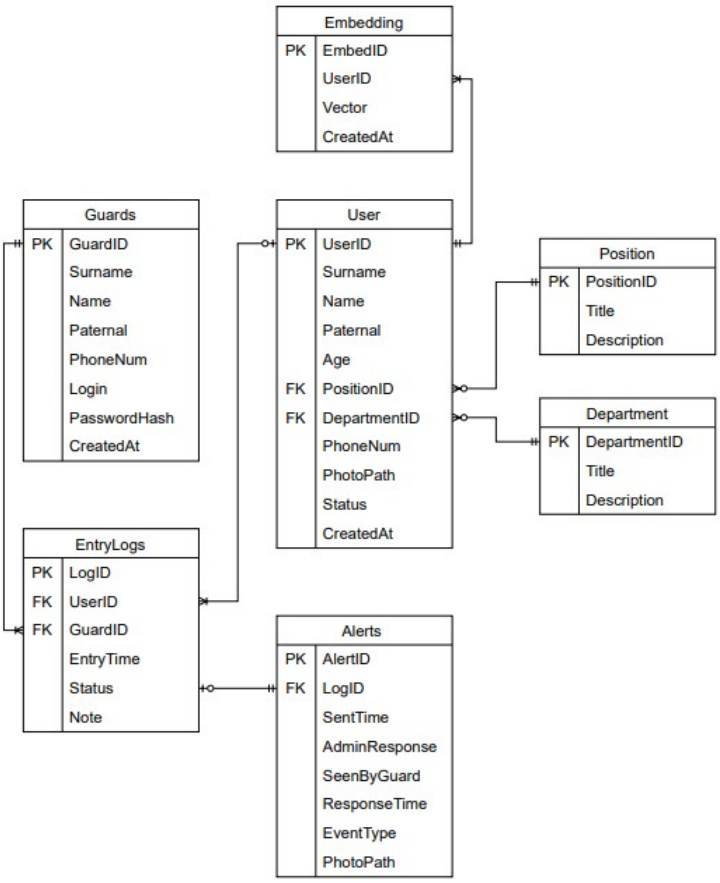

Figure 2: Datalogical database model

The selected database structure is based on the relational model. This is done to ensure the integrity and relationships between tables. This solution also allows you to build complex queries and easily scale the system when expanding the functionality. In addition, using the relational model allows you to organize data in a structured way in the form of tables with clearly defined fields and data types. This ensures ease of navigation, efficiency of query execution and flexibility of information processing, which allows you to easily implement connections between different entities and expand the logic of the system in the future.

For reasons of autonomy and increased security, it was decided to use a local MS SQL Server database. All data is stored locally on the device, eliminating the risks associated with transmitting biometric information over the network or using cloud storage.

For a visual representation of the organization of data in the database and a better understanding of its logic, let's consider the description of the main tables and their fields.

Let's start with the Position table, which stores information about employee job roles and allows you to keep track of structural positions.

Table 1
Position

Attribute naming	Field type	Field size
PositionID	INT (PK)	
Name	NVARCHAR	100
Description	NVARCHAR	255

The Guard table is responsible for representing the administrative level of the system. It stores information about the guards who manage access control to the system.

Table 2
Guard

Attribute naming	Field type	Field size	Limitation
GuardID	INT (PK)		NOT NULL
Surname	NVARCHAR	100	NOT NULL
Name	NVARCHAR	100	NOT NULL
Paternal	NVARCHAR	100	NOT NULL
Phone Num	NVARCHAR	13	NOT NULL
Login	NVARCHAR	50	NOT NULL
Password	NVARCHAR	100	NOT NULL
Salt	NVARCHAR	100	NOT NULL
CreatedAt	DATETIME		

Next we have the table directly with users—(Table 3 Users). It contains data belonging to the system users, its visitors, and employees.

Here, the PositionID field is a foreign key that references the PositionID field of the Position table.

The Embedding table (Table 4) stores the embeddings, key biometric data of individuals, necessary for identification. This table also references the UserID field in the Users table using the UserID field.

Table 3
Users

Attribute naming	Field type	Field size	Limitation
UserID	INT (PK)		NOT NULL
Surname	NVARCHAR	100	NOT NULL
Name	NVARCHAR	100	NOT NULL
Paternal	NVARCHAR	100	NOT NULL
Age	INT		
PositionID	INT (FK)		NOT NULL
DepartmentID	INT (FK)		NOT NULL
PhoneNum	NVARCHAR	13	NOT NULL
PhotoPath	NVARCHAR	255	NOT NULL
Status	NVARCHAR	50	NOT NULL
CreatedAt	DATETIME		

Table 4
Embedding

Attribute naming	Field type	Field size	Limitation
Embed ID	INT (PK)		NOT NULL
UserID	INT (FK)		NOT NULL
Vector	VARBINARY	MAX	NOT NULL
CreatedAt	DATETIME	100	

Also, there is a table EntryLogs (Table 5) in the database. It is designed to keep a log of all access events, tracking each login attempt.

The table is referenced by the UserID field to the UserID field in the Users table, and the GuardID field is a foreign key that references the GuardID field in the Guard table.

Table 5
EntryLogs

Attribute naming	Field type	Field size	Limitation
LogID	INT (PK)		NOT NULL
UserID	INT (FK)		NOT NULL
GuardID	INT (FK)		NOT NULL
EntryTime	DATETIME	100	
Status	NVARCHAR	50	NOT NULL
Note	NVARCHAR	255	NULL

The last table to describe is the Alerts table (Table 6). It contains information about alarming events. The table helps the administration to quickly respond to dangerous situations and keep track of responses.

Table 6
Alerts

Attribute naming	Field type	Field size	Limitation
AlertID	INT (PK)		NOT NULL
LogID	INT (FK)		NOT NULL
SentTime	DATETIME		
AdminResponse	NVARCHAR	50	

Let's present the main relationships in the database in a table, thereby demonstrating the relationship between the system entities:

Table 7
Relationships between database tables

Contact number	Main table	Child table	Communication type
1	Users	Position	M:1
2	Users	Department	M:1
3	Users	Embedding	1:M
4	Users	EntryLogs	1:M
5	EntryLogs	Guard	M:1
6	EntryLogs	Alerts	1:1

To maintain data integrity and logical coherence, the database structure is built using clearly defined relationships between tables. This allows the system to efficiently combine information from different sources and quickly execute complex queries to obtain the necessary data.

3.4. Data management and optimization methods in a high-load personal identification system

The functioning of the automatic face identification system involves processing significant amounts of information in real time. A constant stream of images coming from surveillance cameras generates thousands of data reading, comparison and writing operations every second. Under such conditions, classical approaches to storing and accessing information are not effective enough, which necessitates the use of specialized data management methods focused on speed, scalability and load resistance.

3.4.1. Features of data management in high-load environments

In systems that implement computer vision, the main storage objects are feature vectors (embeddings) obtained from neural networks during facial image processing [37–39]. Each vector is a multidimensional numerical representation of a person's unique features, and the number of such records can reach hundreds of thousands. To ensure operational search and comparison of data, an optimized structure for accessing them is required. Traditional relational storages ensure data integrity and reliability, but at a high frequency of accesses their performance is significantly reduced. Therefore, effective data management in such systems should be based on the principles of multi-tiered storage and load balancing between processing, caching, and indexing modules.

3.4.2. Application of caching technologies

One of the key areas of optimization is the use of a cache for short-term storage of processing results, which allows minimizing the number of repeated requests to the database and speeding up the system in conditions of a large stream of video data. Storing intermediate results in random access memory allows avoiding duplication of calculations and reducing the number of disk operations, which is especially important when processing streams in real time.

Several types of caching can be implemented in the developed system:

- results of the last successful user identifications.
- feature vectors for frames coming from the same camera during a short time interval.
- pre-normalized images ready for comparison.

For this, it is advisable to use high-speed in-memory storages, such as Redis or Memcached, which provide access to data with a delay of less than 1 ms. Redis has the advantage of supporting various data structures (dictionaries, lists, sets) and TTL (Time-to-Live) mechanisms, which allows you to automatically clean up outdated records and maintain the relevance of information. In the system, Redis can also act as a buffer between modules—for example, between image capture and recognition components, ensuring asynchronous data exchange.

To maintain the security of cached data, record lifetime control is implemented (30–60 s), after which they are automatically deleted, which makes it impossible to accumulate personal information. This approach guarantees both increased performance and compliance with confidentiality requirements.

3.4.3. Indexing and optimization of embedding data search

Another important aspect of increasing the efficiency of the system is the indexing of vector data. For fast search of similar faces among a large number of records, traditional B-Tree or Hash indexes are inefficient, therefore specialized structures designed for vector search in multidimensional spaces are used. These include the algorithms FAISS (Facebook AI Similarity Search), Annoy (Approximate Nearest Neighbors Oh Yeah) and HNSW (Hierarchical Navigable Small World), which provide the search of the nearest neighbors (Nearest Neighbor Search) with sublogarithmic complexity.

The implementation of vector indexes allows you to search for identical or similar faces even with a large number of records (over 1 million vectors) with a response time of less than 50 ms. In the proposed architecture, it is advisable to use a database with support for vector types, such as PostgreSQL with the pgvector extension, or specialized repositories—Milvus or Weaviate, which are focused on high-speed search of embedding vectors.

3.4.4. Scaling and load distribution

To ensure stable operation of the system with an increase in the number of connected cameras and users, the principle of distributed architecture is used, which allows you to effectively scale system

components depending on the type of load. This approach provides the ability to simultaneously process dozens of video streams in real time without reducing performance or recognition accuracy.

The main modules of the system—image processing, anti-spoofing, identification, cache and database—can be implemented as separate microservices that interact via API or messaging system). This allows you to perform stream processing in parallel, as well as share resources between services depending on their functional role. For example, one node can process video from high-resolution cameras, while another—serves requests for identification or analytics.
In such an architecture, scaling can be done in two ways:

- vertical scaling—increasing the resources of a single node (CPU, GPU, RAM) to process more complex recognition models;
- horizontal scaling—adding new nodes or containers that are automatically connected to the system using orchestration mechanisms.

The use of containerization allows you to quickly deploy new instances of services depending on the current load. For example, if the number of active cameras increases over a certain period of time, the system can automatically create additional copies of the identification service to maintain constant throughput.

In addition, load balancing between computing nodes ensures an even distribution of recognition requests, preventing overloading of individual servers. For this, both classic HTTP-level solutions and specialized tools with dynamic routing control can be used.

An important element of the architecture is also monitoring the performance and status of components, which can be implemented through the Prometheus or Grafana systems. This allows you to track processing delays, resource usage, and the number of active threads in real time, which is critical for maintaining the stability of a highly loaded system.

Thanks to the implementation of these approaches, the system remains scalable, fault-tolerant and adaptive to load changes, maintaining minimal response time even during peak requests.

3.4.5. Log management and aggregated tables

To store a large volume of access logs and events and at the same time ensure fast analytical queries, the principle of aggregated (summary) tables is used. Regular data aggregation by time intervals (hour, day, week) allows you to significantly reduce the execution time of administrator queries without losing the accuracy of statistics. Additional use of indexes on the UserID, CreatedAt, Status fields provides search optimization by the most commonly used criteria.

Implementation of the proposed approaches—caching, vector indexing and distributed data management—allows you to significantly increase the speed of the personal identification system without losing the accuracy of recognition. Thus, the system is transformed into a highly loaded information platform capable of scalability, fault tolerance and effective data flow management, hich fully meets the modern requirements for computer vision-based security system architectures.

4. Accuracy and completeness of the developed system

In pattern recognition, information retrieval, and classification, precision [39] is the proportion of relevant samples among those found, while recall [40] is the proportion of the total number of positive samples that were actually found. Both precision and recall are therefore based on an understanding and measure of relevance. Precision should not be confused with accuracy, which is the proportion of correctly predicted results, both positive and negative [40]. Precision only applies to positive results.

TP = True Positive, i. e. when the actual value was “yes”, the model predicted “yes” (i. e. correct prediction = correctly worked)

FP = False Positive, i. e. when the actual value was “no”, the model predicted “yes” (i. e. incorrect prediction = falsely worked)

TN = True Negative, i. e. when the actual value was “no”, the model predicted “no” (i. e. correct prediction)

FN = False Negative, i. e. when the actual value was “yes”, the model predicted “no” (i. e. incorrect prediction)

Accuracy is the proportion of all correctly classified data (positive and negative results). Mathematically, it is expressed as:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

That is, it is the ratio of the number of correctly predicted cases to the total number of cases. An ideal model would have zero false positives and zero false negatives, and therefore the accuracy would be 1.0, or 100%.

Since it includes all four outputs from the error matrix (TP, FP, TN, FN), given a balanced dataset with the same number of examples in both classes, accuracy can serve as a general indicator of the quality of the model. For this reason, accuracy is often the default evaluation metric for typical models or those that do not perform special tasks.

The true positive rate (TPR), or the proportion of all actual positive outcomes that are correctly classified as positive, is also called completeness.

Mathematically, completeness is expressed as [41]:

$$Recall = \frac{TP}{TP + FN}$$

and describes how many objects the algorithm finds that are true anomalies—the percentage of true anomalies among all those that the system has identified as anomalies. The higher the accuracy, the fewer false positives, the higher the completeness, the fewer false positives. By changing the algorithm parameters, you can build a curve of the dependence of accuracy and completeness

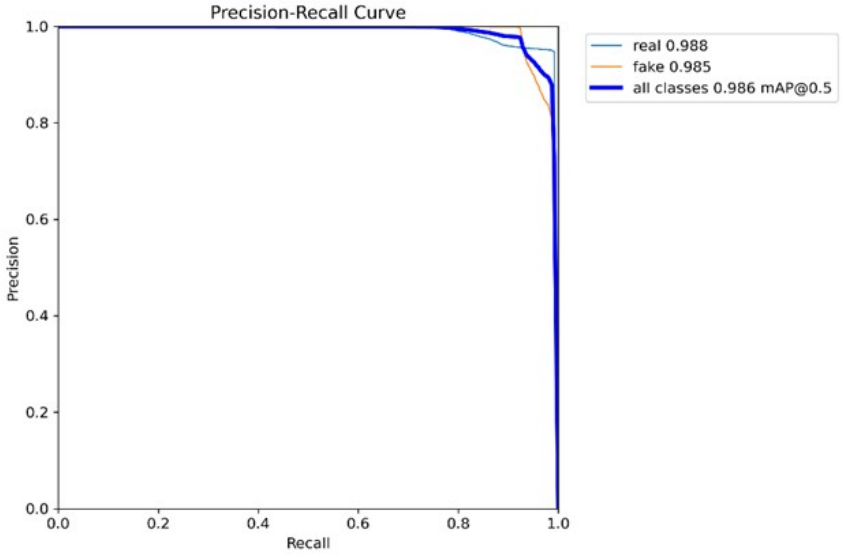

Figure 3: Accuracy-completeness curves of the developed system

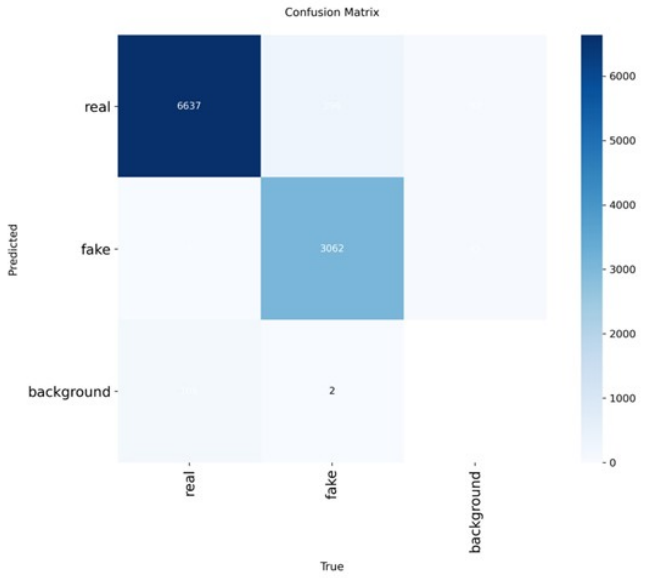

Figure 4: Inconsistency matrix

Accuracy is the proportion of all results that the model classifies as positive and which are really such. Mathematically, it is expressed as follows:

$$Precision = \frac{TP}{TP + FP}$$

and formally describes the confidence level of the classifier—the percentage of outliers found that are actually outliers.

5. Conclusion

The designed database structure demonstrates a thorough approach to ensuring both effective data organization and security. The described clearly defined relationships between tables guarantee data structure, where each record is logically linked to the corresponding entities, ensuring information integrity. Flexibility in forming complex queries to obtain the necessary data and ensuring system scalability is obtained. This allows adding new tables or fields without global restructuring of the schema. A well-thought-out relationship system is a guarantee of data integrity and correctness even with increasing load and number of users. In addition to the structure, special attention is paid to data security, which is critically important for a personal identification system. Transactional support guarantees the integrity of records during update and insert operations. A flexible audit system has also been implemented, which allows tracking all access attempts and administrative actions, which provides monitoring of suspicious activity and the possibility of post-disaster analysis. No less important is the protection of security guards' accounts by storing hashed passwords. This measure protects the system from unauthorized login attempts. Thus, all these mechanisms together form a reliable and secure basis for the functioning of the personal identification system.

The results obtained confirmed the correctness of the chosen approach: the system correctly identifies users even on weak cameras and provides protection against spoofing.

Declaration on Generative AI

While preparing this work, the authors used the AI programs Grammarly Pro to correct text grammar and Strike Plagiarism to search for possible plagiarism. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the publication's content.

References

[1] P. Petriv, I. Opirskyy, N. Mazur, Modern Technologies of Decentralized Databases, Authentication, and Authorization Methods, in: Cybersecurity Providing in Information and Telecommunication Systems II, vol. 3826, 2024, 60–71.

[2] N. Ghedia, C. Vithalani, A. M. Kothari, R. M. Thanki, Moving Objects Detection using Machine Learning, Springer Int. Publ., Switzerland, 2022, 65–68.

[3] N. Xu, W. Lin, X. Lu, Y. Wei, Video Object Tracking: Tasks, Datasets, and Methods, 2024, 75–79.

[4] P. Skladannyi, et al., Adaptive Methods for Embedding Digital Watermarks to Protect Audio and Video Images in Information and Communication Systems, in: Classic, Quantum, and Post-Quantum Cryptography (CQPC), vol. 4016 (2025) 13–31.

[5] F. Hassan, M. H. I. Ishak, M. S. M. Ali, M. A. M. Basri, M. S. A. Mahmud, N. Sunar, Methods and Applications for Modeling and Simulation of Complex Systems, in: 22nd Asia Simulation Conf. (AsiaSim 2023), Langkawi, Malaysia, Proc. Part I, Springer, 2023, 24.

[6] G. J. Edwards, T. F. Cootes, C. J. Taylor, Face Recognition using Active Appearance Models, in: Comput. Vis., Lect. Notes Comput. Sci., vol. 1407, 2006, 581–595.

[7] S. Sankaranarayanan, A. Alavi, C. Castillo, R. Chellappa, Triplet Probabilistic Embedding for Face Verification and Clustering, in: IEEE Int. Conf. Biometrics Theory, Appl. Syst. (BTAS), 2016, 1–8.

[8] B. Zhurakovskyi, et al., Modifications of the Correlation Method of Face Detection in Biometric Identification Systems, in: Cybersecurity Providing in Information and Telecommunication Systems (CPITS 2022), vol. 3288, 2022, 55–63.

[9] B. Zhurakovskyi, et al., Traffic Control System based on Neural Network, in: Digital Ecosystems: Interconnecting Advanced Networks with AI Applications (TCSET 2024), Lect. Notes Electr. Eng., vol. 1198, 2024, 522–542. doi:10.1007/978-3-031-61221-3_25

[10] N. Fedorova, et al., Software System for Processing and Visualization of Big Data Arrays, in: Adv. Comput. Sci. Eng. Educ. (ICCSEEA 2022), Lect. Notes Data Eng. Commun. Technol., vol. 134, 2022, 324–336. doi:10.1007/978-3-031-04812-8_28

[11] B. Zhurakovskyi, et al., Machine Learning-based Environmental Monitoring and Analysis System, in: Cybersecurity Providing in Information and Telecommunication Systems (CPITS 2025), vol. 3991, 2025, 183–203.

[12] J. Zhu, et al., Online Multi-Object Tracking with Dual Matching Attention Networks, 2018.

[13] T.-Y. Lin, et al., Focal Loss for Dense Object Detection, 2018.

[14] I. Liminovych, et al., Protection System for Analysis of External Link Placing, in: Cybersecurity Providing in Information and Telecommunication Systems (CPITS 2024), vol. 3654, 2024, 179–188.

[15] D. Virovets, et al., Integration of Smart Contracts and Artificial Intelligence using Cryptographic Oracles, in: Classic, Quantum, and Post-Quantum Cryptography, vol. 3829, 2024, 39–46.

[16] X. Yang, J. Yan, Arbitrary-oriented Object Detection with Circular Smooth Label, in: Comput. Vis. – ECCV 2020, Cham: Springer, 2020, 677–694.

[17] B. Zhurakovskyi, et al., Calculation of Quality Indicators of the Future Multiservice Network, in: Future Intent-Based Networking, Lect. Notes Electr. Eng., vol. 831, 2022, 197–209. doi:10.1007/978-3-030-92435-5_11

[18] A. Bedagkar-Gala, S. K. Shah, A Survey of Approaches and Trends in Person Re-Identification, 2014.

[19] B. Zhurakovskyi, et al., Data Protection in the Automated Agribusiness Management System, in: Cybersecurity Providing in Information and Telecommunication Systems II (CPITS-II 2024), vol. 3826, 2024, 267–275.

[20] B. Zhurakovskyi, et al., Enhancing Information Transmission Security with Stochastic Codes, in: Classic, Quantum, and Post-Quantum Cryptography, vol. 3829, 2024, 62–69.

[21] B. Zhurakovskyi, et al., Comparative Analysis of Modern Formats of Lossy Audio Compression, in: 1st Int. Conf. on Cyber Hygiene and Conflict Management in Global Information Networks, vol. 2654, 2020.
[22] L. Shapiro, G. Stockman, Computer Vision, Prentice-Hall, Upper Saddle River, NJ, 2000.
[23] A. D. Vairamani, A. Nayyar, A. Kumar, R. Jain, Object Tracking Technology: Trends, Challenges and Applications, Springer Nature, Singapore, 2023, 1–23. doi:10.1007/978-981-99-3288-7
[24] D. A. Forsyth, J. Ponce, Computer Vision: A Modern Approach, Prentice-Hall, Upper Saddle River, NJ, 2002.
[25] M. Everingham, et al., The PASCAL Visual Object Classes (VOC) challenge, 2009.
[26] H. Wang, X. Zhu, S. Gong, T. Xiang, Person Re-Identification in Identity Regression Space, 2018.
[27] A. Chung, S. Kim, E. Kwok, M. Ryan, E. Tan, R. Gamadia, Cloud Computed Machine Learning-based Real-Time Litter Detection using Micro-UAV Surveillance, in: IEEE MIT Undergraduate Research Technol. Conf. (URTC 2018). doi:10.1109/URTC45901.2018.9244800
[28] X. Zhang, et al., AlignedReID: Surpassing Human-Level Performance in Person Re-Identification, 2017. doi:10.48550/arXiv.1711.08184
[29] B. Zhurakovskyi, et al., Processing and Analyzing Images based on a Neural Network, in: Cybersecurity Providing in Information and Telecommunication Systems, vol. 3654, 2024, 125–136.
[30] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR 2016), 2016. http://surl.li/gppvv
[31] Z. Zou, Z. Shi, Y. Guo, J. Ye, Object Detection in 20 Years: A Survey, 2019. doi:10.48550/arXiv.1905.05055
[32] MediaPipe Documentation. https://developers.google.com/mediapipe
[33] C. Gouider, H. Seddik, YOLOv4 Enhancement with Efficient Channel Recalibration Approach in CSPDarknet53, in: IEEE Inf. Technol. Smart Ind. Syst. (ITSIS 2022), 2022, 1–6.
[34] Q. W. et al., Learning Modulated Loss for Rotated Object Detection, in: AAAI Conf. Artif. Intell., 35(3), 2021, 2458–2466.
[35] B. Zhurakovskyi, et al., Using the Latest Methods of Cluster Analysis to Identify Similar Profiles in Leading Social Networks, in: Inf. Technol. Implement., vol. 3646, 2023, 116–126.
[36] J. Dai, Y. Li, K. He, J. Sun, R-FCN: Object Detection via Region-based Fully Convolutional Networks, in: Adv. Neural Inf. Process. Syst. (NIPS 2016), 2016, 379–387.
[37] V. Dudykevych, et al., Detecting Deepfake Modifications of Biometric Images using Neural Networks, in: Cybersecurity Providing in Information and Telecommunication Systems, CPITS, vol. 3654 (2024) 391–397.
[38] Y. Kostiuk, et al., Application of Statistical and Neural Network Algorithms in Steganographic Synthesis and Analysis of Hidden Information in Audio and Graphic Files, in: Classic, Quantum, and Post-Quantum Cryptography (CQPC), vol. 4016 (2025) 45–65.
[39] D. L. Olson, D. Delen, Advanced Data Mining Techniques, Springer, 1st ed., 2008, 138.
[40] J. P. Mower, PREP-Mt: Predictive RNA Editor for Plant Mitochondrial Genes, BMC Bioinf., 6 (2005) 96. doi:10.1186/1471-2105-6-96
[41] T. Saito, M. Rehmsmeier, G. Brock, The Precision-Recall Plot is More Informative than the ROC Plot when Evaluating Binary Classifiers on Imbalanced Datasets, PLoS ONE, 10(3) (2015). doi:10.1371/journal.pone.0118432