Enhancement of the Facial Recognition Module in the “Safe University” System through Adaptive Fine-Tuning

Natalya Denissova, Irina Dyomina*, Aizhan Tlebaldinova, and Kurmash Apayev

Department of Information Technologies, D. Serikbayev East Kazakhstan technical university, Oskemen, Kazakhstan
Email: ndenisova@edu.ektu.kz (N.D.); idyomina@edu.ektu.kz (I.D.); atlebaldinova@edu.ektu.kz (A.T.); kapaev@edu.ektu.kz (K.A.)
*Corresponding author

***Abstract*—This article explores methods for improving the quality of existing facial biometric recognition systems by fine-tuning the model on new data. It examines the overall framework reflecting the fundamental operating principle of the biometric identification security system, as well as the main approaches and methods for addressing this task using the Deep Neural Network (DNN) face detection method in OpenCV. A facial recognition software suite has been developed, which includes: a detection module, a head position determination module, a user identification module, an Access Control and Management System (ACMS) module, and a training module. Research on existing methods to enhance the accuracy of identification algorithms and systems has been conducted. An analysis of the increase in recognition rates after system fine-tuning for different times of day was performed. The results of the study showed that the developed module ensures high accuracy and reliability. The recognition rate increased by approximately 4–5% as a result of system fine-tuning. Additionally, it is worth noting that ACMS with facial recognition technology represents a powerful tool for educational institutions seeking to automate their attendance tracking processes. This step marks significant progress in applying advanced technologies to increase the efficiency and accuracy of attendance management.**

***Keywords*—identification, recognition systems, recognition algorithms, Deep Neural Network (DNN) face detection method in OpenCV, fine-tuning, integration with Access Control and Management System (ACMS)**

I. Introduction

In recent years, facial recognition has emerged as a key technology in the realm of security and access control, offering reliable and contactless methods for identity verification. In university campuses, where both security and convenience are paramount, facial recognition systems can significantly enhance protection by providing automatic building access and monitoring the presence of students and staff. However, despite advancements in this field, many facial recognition systems face challenges related to accuracy and reliability, particularly under varying lighting conditions, differences in poses and facial expressions, and image noise [1–3]. The aim of this article is to explore methods for improving the quality of existing facial recognition systems through fine-tuning the model on new data. Fine-tuning allows an existing model to be adapted to the specific conditions and requirements of the university, enhancing its ability to distinguish faces in real-world scenarios. Introducing fine-tuning into an existing facial recognition system for university security can significantly boost its accuracy and reliability, thereby enhancing overall campus safety. This article discusses the process of model fine-tuning, from data preparation to integrating the updated model into the existing infrastructure. Special attention is given to methods of data collection and processing to ensure high-quality training datasets, as well as to the analysis of results before and after fine-tuning. The novelty of this research lies in the development of a new methodology for training and fine-tuning a facial recognition model integrated with an Access Control and Management System (ACMS). This methodology includes a training module and an image database, updating the database and fine-tuning the model, and setting and subsequently adjusting the recognition threshold. This article presents both the theoretical foundations and practical aspects of implementing an enhanced facial recognition system. We also discuss potential challenges and propose solutions to overcome them, which will assist other educational institutions and organizations in adapting and improving their security systems.

II. Literature Review

Currently, the security of educational institutions plays a crucial role for several reasons: protecting students and staff, countering threats, maintaining reputation, and creating a conducive educational environment. With the advancement of technologies, the implementation of automated access control and management systems is becoming increasingly sought after. Many automated access control systems are being introduced, such as biometric methods and fingerprint-based access systems,

Manuscript received July 10, 2024; revised August 26, 2024; accepted September 19, 2024; published January 23, 2025.

doi: 10.12720/jait.16.1.144-155

but these existing systems face numerous complexities that lead to time consumption. Thus, the implementation of facial recognition systems appears to be a more promising solution in this area.

Rameswari *et al.* [4] discuss an efficient and modern algorithm called FaceNet. The access system using this algorithm employs face encoding to detect faces and eyes, which works effectively under varying light conditions.

Facial recognition in attendance management systems represents an advanced technological solution aimed at improving and optimizing the attendance tracking process in various educational institutions and organizations. Jha *et al.* [5] propose a system that utilizes Haar cascade algorithms for face detection and recognition. Face detection algorithms identify and extract facial regions from input images or video frames, isolating the necessary facial details for further analysis. Subsequently, Local Binary Patterns Histograms (LBPH) face recognition algorithms compare these features with pre-registered faces stored in the system's database, calculating confidence scores to determine the individual's identity.

The system has a user-friendly interface, allowing administrators to easily manage attendance records. They can easily add or remove students from the system's database, access attendance reports, and track attendance data in real-time. The proposed attendance management system revolutionizes the traditional attendance tracking process by offering enhanced accuracy, efficiency, and security, as well as providing real-time monitoring and comprehensive attendance reporting capabilities [5, 6].

Abdulabas and Al-Shakarchy [7] propose a facial identification model based on a Deep Neural Network (DNN). The proposed method extracts spatial information available in the image, analyzes this information to derive essential features, and makes identification decisions based on these features. This model demonstrates successful and promising results, with an accuracy of 99.5% (±0.16%) and a loss function value of 0.0308 on the Pins face recognition dataset for identifying 105 subjects.

Many researchers are studying the process of biometric-based recognition, making it a current and very complex task. The recognition stages for various systems include: scanning the input image (obtained from various sources) for human faces [8, 9], image segmentation (highlighting or defining faces) [10–13], and recognition (applying various computer vision algorithms) [14–22].

Each stage of recognition is currently being deeply studied by researchers. For example, Wang [13] addresses the problem of object detection and segmentation from video streams and individual images. The work presents image classification based on the scene (environment of the recognizable object), semantic segmentation, face analysis, and group analysis. Recognizable images are represented pixel-by-pixel, and Convolutional Neural Networks (CNNs) and highly efficient forward and backward propagation algorithms are applied for their recognition.

The segmentation stage is also a very complex task. Computer vision allows for object detection and recognition. Facial recognition is based on matching and combining sets of key points, forming a set of classifiers [14]. Approximately 6000 classifiers are required for overall face recognition (each classifier corresponds to a specific facial feature), making the task of computer vision very "costly." Many researchers aim to simplify and ease the recognition process. In the work by Hiramatsu *et al.* from Brazil [15], a method for face detection and tracking using a single camera is presented. Based on a matrix of classifiers, a cascade selection of the most "significant" classifiers was performed, reducing the number of iterations and increasing the performance of real-time face detection from a video stream.

Many researchers use the technique of comparing a face to a "template". Robertson *et al.* [16] show that Facial Recognition Systems (FRS) use template matching to compare two faces and generate a similarity score reflecting the degree of their resemblance. More modern recognition methods involve Convolutional Neural Networks (CNN), but issues arise even with this approach. It is generally assumed that CNNs are invariant to small image transformations, either due to their convolutional architecture or because they have been trained using data augmentation (artificial generation of new data based on existing ones). Azulay and Weiss [17] demonstrated that the convolutional architecture does not ensure invariance, as it ignores the classic sampling theorem, and data augmentation does not provide invariance because CNNs learn to be invariant only to transformations for images that are very similar to typical images from the training set. The authors presented two possible solutions to this problem: smoothing intermediate representations and increasing the data volume, showing that they provide only partial solutions at best.

Typically, a facial recognition system is a software-hardware complex for automatic verification or authentication of identity using a digital image. When solving the problem of facial recognition on two-dimensional images, another important factor plays a role —the angles of facial projection that arise during photographing [18].

Sarraf and Kabia [19] showed that improved feature extraction from images is achieved using an "attention mechanism," which allows computer vision models to extract essential features related to the type of object location, resulting in improved recognition quality.

Another solution to the recognition problem is to represent images as evenly spaced pixel arrays and convolve highly localized elements [20]. In this work, images are presented as semantic visual tokens, with transformers launched for dense modeling of token relationships during image processing, enabling the sensible processing of various parts of the image depending on the "scene".

He *et al.* [21] proposed the Region Generation and Assessment Network (RGANet) for effective human body region detection and important area highlighting. In the proposed RGANet, the authors developed a Region Generation Module (RGM) that uses a pre-trained Contrastive Language-Image Pre-training (CLIP) to determine human body regions' locations using semantic prototypes extracted from textual descriptions. Serbetci and Akgul [22] use Binary Neural Networks (BNN) to learn binary hash codes for efficient person Re-Identification (ReID). In their research, the authors proposed solving the gradient mismatch problem by developing a multi-branch ensemble model consisting of multiple weakly learned hash codes. Specifically, the authors' proposed structure combines gradients from multiple branches, allowing for better gradient approximation and network ordering. By combining the efficiency of BNN and hash code learning, the authors achieved an efficient ensemble model that is effective in both feature extraction and ranking stages, showing that the proposed model outperforms the traditional BNN ensemble by more than 7%, being nearly 10 and 2 times more efficient in terms of CPU consumption and memory volume, respectively.

Thus, many researchers study recognition issues, but there are still insufficient studies that consider the problem comprehensively for a specific security system or corporate information system. This work examines the functioning of a facial recognition system from building a person's template based on existing biometric characteristics, fine-tuning, and ultimately updating the template for recognition and integrating the recognition module with the institution's security system and the entire corporate system of the institution.

III. Materials and Methods

The overall functioning scheme of the "Safe University" system is presented in Fig. 1.

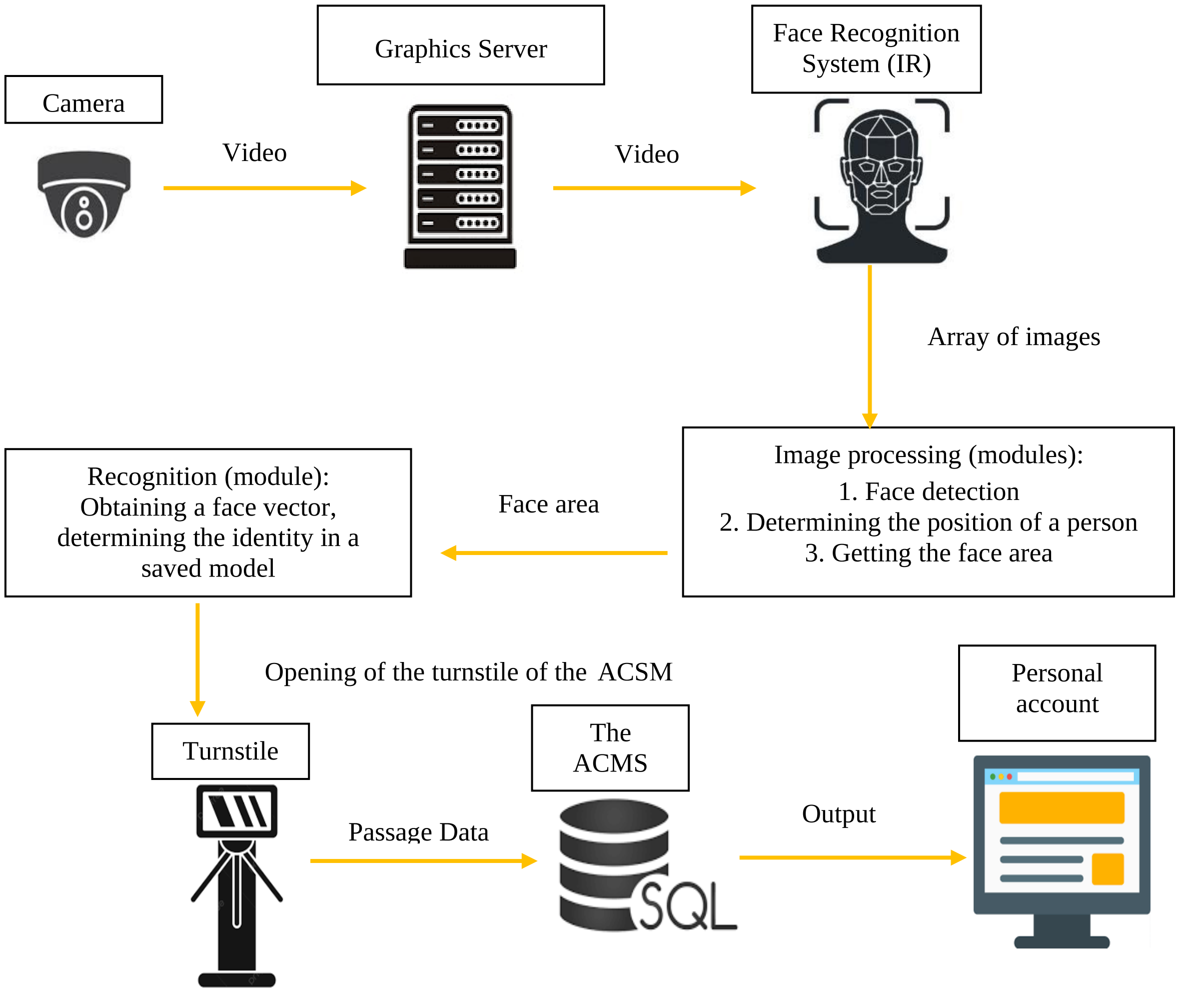


Fig. 1. Operational diagram of the "Safe University" system.

The implemented system, based on the given scheme, uses a regular IP camera with a frame rate of up to 24 frames per second, a resolution of 1920×1080 pixels (Full HD), and a fixed focus (stationary). This is standard equipment capable of transmitting sufficiently high-quality images for most facial recognition tasks. A stationary camera can provide detailed images if it is properly installed and configured. Key factors include lighting, distance to the subject, viewing angle, and resolution. In your case, with a resolution of 1920×1080, the camera will be able to capture faces at a certain distance with enough detail for recognition. The closer the subject is to the camera and the better the lighting, the more details will be captured in the image.

According to the provided scheme, the image array is fed into the facial recognition module (IR), where detection, head position determination, face area extraction, and recognition are performed. The image array is a continuous stream of data (images) from the camera. According to the scheme, the original image is sent to the facial recognition module (IR), where detection takes place. This model is only used in indoor environments, for access control systems with turnstile throughput of up to 2 seconds.

According to the presented diagram, the initial image is fed into the face recognition module (IR), where detection is performed. The face detection methods used include: the Haar cascade face detector in OpenCV [23], the DNN face detector in OpenCV [24–26], the HoG face detector in Dlib [27, 28], and the deep learning-based face detector in Dlib (the Max-Margin Object Detection (MMOD) method) [29, 30].

As a result of testing the presented methods, based on the available images, the DNN detector demonstrated an advantage, as it showed the highest accuracy, noise resistance, and the lowest time costs.

Next, head position determination methods and face vector formation methods are applied for identification. All candidates who pass identification are valid for the ACMS system and gain access, but for security systems, it is important that the recognition rate is sufficient for reliable identification. In the described system, the minimum recognition rate sufficient for identification is 69%.

Our research explores the possibility of system fine-tuning when the initial data quality is poor (e.g., blur, noise, opacity, etc.). The addition of photos is carried out from images obtained during identification in the system, or the system operator can upload images, which increases the recognition rate.

Upon successful identification, data about passages are recorded in the ACMS, and data on how identification occurred (via access card or IR system) are saved. For convenience, passage data is displayed in the employee’s personal account, and data on department employees is displayed in the head of the structural department’s personal account.

Let us consider in more detail how the face recognition module (IR) works.

Facial recognition using DNN involves several stages, starting from image preprocessing to face classification or verification. In this work, the DNN method is considered a key approach to improving the accuracy and reliability of the facial recognition system within a university campus. DNN effectively adapts the model to specific conditions, such as changing lighting, various facial poses, and expressions, due to its ability to learn from large datasets and identify complex patterns [31]. It provides high processing speed and recognition accuracy, which is critical for real-time tasks such as access control and attendance monitoring [1]. The choice of DNN is justified by its adaptability, noise resistance, and ability to be retrained on new data, allowing the “Safe University” system to remain reliable and efficient even in changing conditions.

The main stages of face recognition using DNN are presented in Fig. 2.

During the image preprocessing stage, normalization and data augmentation are performed. The normalization step includes resizing images, aligning them to the center of the face, and normalizing pixel values. Data augmentation is used to increase the diversity of the training set (through rotations, reflections, brightness and contrast adjustments).

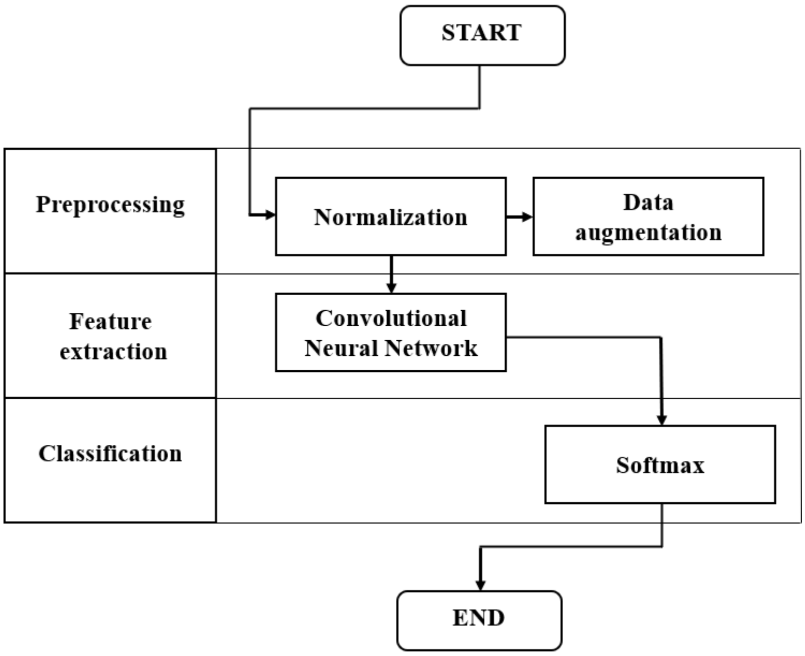


Fig. 2. The flowchart of the proposed DNN model.

The next stage, feature extraction of the facial image, is performed using a Convolutional Neural Network (CNN). Convolutional layers apply filters to the image to highlight features such as edges, corners, and textures.

Let III be the input image, KKK be the filter, and *O(i,j)O(i,j)O(i,j)* be the output element, which is computed as:

$$O(i,j) = (I \cdot K)(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(i+m, j+n) \cdot K(m,n) \quad (1)$$

where *M*, *N*-filter sizes [32].

As an activation function, ReLU (Rectified Linear Unit) [33] is used, which adds nonlinearity to the model and helps in learning more complex dependencies.

The next stage is the pooling layer, which reduces the dimensionality of the data while preserving important features. For example, max pooling selects the maximum value in each subregion of the image.

The ReLU function is defined as [34]:

$$O(i,j) = max_{m,n} I(i+m, j+n) \quad (2)$$

After several such layers (convolutional, activation, and pooling), the data passes through a fully connected layer. Fully connected layers integrate all extracted features, and each neuron in this layer is connected to all neurons in the previous layer, allowing the model to make decisions based on all available data [35].

Finally, the result is processed in the output layer, which provides the final classification result. For the face recognition task, this can be either face identification or verification, determining whether the given face matches a known face in the database. The output layer typically uses the softmax activation function for classification [32]:

$$\hat{y_i} = \frac{e^{z_i}}{\sum_j e^{z_j}} \quad (3)$$

where y_i—predicted class probabilityi, z_i—assessment value. During the training of the model, the backpropagation algorithm is used to minimize the loss function, such as cross-entropy [36]:

$$L(y, \hat{y}) = -\sum_i y_i log\ (y_i) \quad (4)$$

where *y*—represents the true labels and ŷ—represents the predicted probabilities. Finally, the gradients of the loss function with respect to the weights and biases are computed and updated using the gradient descent method.

Thus, the cross-entropy loss function is used for training, and its minimization is carried out through backpropagation and gradient descent, allowing the DNN model to effectively classify the input data. During system fine-tuning, face vectors obtained from processing new face images are added to the database. The following methods are employed: head pose estimation and Euclidean distance method.

The head pose estimation method involves several stages. First, the input image is converted to grayscale for simplification, then faces are detected in the image using a face detector. For each detected face, a key point predictor identifies characteristic points such as the corners of the eyes and mouth, the tip of the nose, and the chin. These points are used to solve the head pose estimation problem using the cv2.solvePnP function [37], which computes the rotation vector and translation vector. The obtained data is converted into Euler angles (pitch, yaw, and roll), characterizing the head orientation relative to the camera.

Fig. 3 illustrates the coordinate system used in the pose estimation process. The camera's X-axis points to the right, the Y-axis points downward, and the Z-axis points forward.

By integrating these steps, the system can accurately determine the orientation of a person's head relative to the camera, enhancing the robustness and accuracy of the facial recognition process.

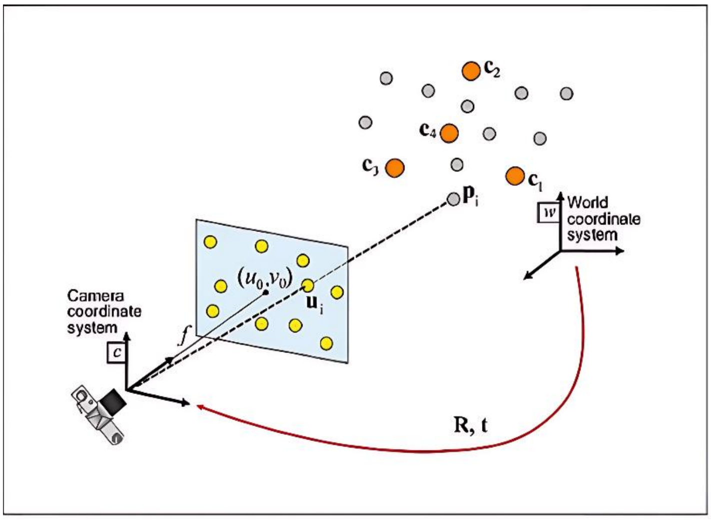


Fig. 3. Determining head position [37].

The points expressed in the world coordinate system X_w are projected onto the image plane (u, v) using the perspective projection model Π and the camera's intrinsic parameter matrix.

In mathematical terms, the relationship between the 3D world coordinates and the 2D image coordinates can be expressed as [37]:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A\Pi^{c}T_{w}\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

The estimated pose, therefore, consists of the rotation vectors (rvec) and translation vectors (tvec), which allow transforming a three-dimensional point, expressed in the world frame, into the camera frame:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} r_{12} r_{13} t_x \\ r_{21} r_{22} r_{23} t_y \\ r_{31} r_{32} r_{33} t_z \\ 0\ \ 0\ \ 0\ \ 1 \end{bmatrix}\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (6)$$

Next, face vectors are formed and identity is determined based on the Euclidean distance method [38–40]. The Euclidean distance method is used to measure the "straight-line" distance between two points in a multidimensional space. In two-dimensional space, this distance can be represented as the length of the segment connecting the two points. The rule for this method is as follows: the closer the distance between two points, the higher the likelihood that these points belong to the same class or cluster, and conversely, the greater the distance, the lower this likelihood.

To determine the recognition accuracy percentage, a simple approach is used. A threshold value of Euclidean distance is set, below which two points are considered to belong to the same class. Then, the proportion of correctly classified points relative to the total number of points is calculated and multiplied by 100%, determining the recognition accuracy percentage.

All these methods are considered for integration with a turnstile system. In our case, we consider the Perco-S-20 access control system. Data from the Perco-S-20 database is used for identification and integrated with the corporate system, particularly with the Employee Time Tracking and Student Attendance System (Fig. 1). Turnstile systems with facial recognition terminals represent an effective means of access control. Their operation is based on a simple yet effective principle: facial images are entered into a database and linked to the corresponding user accounts. Additionally, all data from the turnstile system is transmitted to the corporate system and user personal accounts. The functioning of the facial recognition system begins with building a person's template based on the available biometric samples, most commonly using two-dimensional images or unpacked frames from a video stream [30].

In this article, a dataset obtained under certain conditions is used to solve the face recognition problem. Primary facial images were collected considering the factors presented in Table I.

TABLE I. FACTORS AND THEIR PERMISSIBLE VALUES

Factor	Value	
	Minimum value	Maximum value
Distance to head it is determined by the technical specifications of the camera	0.5 m	4 m
Head angle along all axe	0%	10–15%
Lighting	250 lux	No limit
Camera resolution	600×800 px	1920×1800 px

All images were pre-annotated, meaning each image was assigned labels that allow the facial recognition system to identify key features of the given image used during training. The annotation includes marking key facial points and defining its boundaries [41, 42].

IV. RESULT AND DISCUSSION

To deploy the Face ID recognition system, a comprehensive infrastructure was prepared, which includes: server equipment, initial employee identification points, the Perco-S-20 access control system, cameras integrated with the ACMS, and the developed software suite (Table II).

TABLE II. DESCRIPTION OF FACE ID RECOGNITION SYSTEM COMPONENTS

№	Name	Description
1	Access Control System equipment (ACS)	Camera specifications: Video camera type: dome, Standard: HD-CVI, Maximum resolution: 1920x1080, 2.0 MP, Matrix type: CMOS 1/2.7, Min. focal length: 2.8 mm, Lens: fixed focus, Video codec support: H.264, H.264+, H.265, H.265+. Power type: PoE, from an adapter. Turnstile specifications: Model PERCo-CT01 is a three-bar stainless steel turnstile with a compact design, designed for access control. Supports one-way and two-way passage, operates in standalone mode or as part of an access control system. It has protection against unauthorized access and automatic unlocking in emergency situations. ACS characteristics: PERCo-S-20 - hardware and software access control system to ensure security at enterprises, offices, educational institutions and other facilities. Includes controllers, readers, electromechanical locks, turnstiles and software for access control, timekeeping and event monitoring, Database: Firebase
2	Server equipment	Server characteristics: AMD EPYC 7452 32-Core Processor, Video adapter: RTX 2070 8GB, SSD 1Tb. The server has the necessary software for image processing and analysis installed Ubuntu Linux (64-bit), OpenCv 4.5, Nvidia Drivers for Cuda 10
3	Points of primary identification of employees	Primary identification points are created to form a person template; each point is equipped with a camera and a TV (Fig. 4). The camera works in real time to fill the database (DB) with a set of 64 photos for further recognition
4	Software package	The software includes facial recognition algorithms trained on constantly updated images of visitors

We have developed a multi-threaded facial recognition system (IR) based on several recognition algorithms: detection algorithms, head pose estimation algorithms, and image lighting analysis algorithms. These algorithms ensure high accuracy (from 60% to 96%) and recognition speed (from 70 ms to 125 ms) and can adapt to various lighting conditions and camera angles. The permissible head position (tilt angle) is 10% both vertically and horizontally. If the head position exceeds this threshold, the face does not pass the recognition module and is ignored. Additionally, minimum lighting parameters must be met - in our case, artificial lighting is used.

During testing, approximate lighting boundaries were determined, more than 300 lux. If the illumination is below this threshold, the face detection model may work less effectively or even stop working altogether due to insufficient light for proper face recognition.

The developed facial recognition software suite consists of: a detection module, a head pose estimation module, a user identification module, an Access Control and Management System (ACMS) module, and a training module. A client-side web application has also been developed.

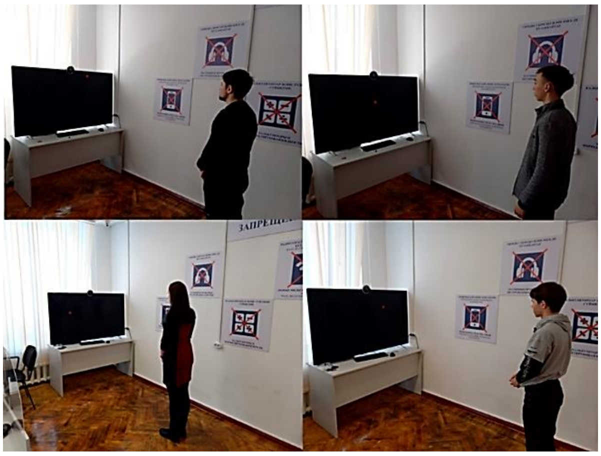


Fig. 4. Primary identification point.

- Detection Module: This module searches for the nearest face, determines the distance to the face, and captures the Region of Interest (ROI). In Algorithm 1, a fragment of the FaceService file (face detection and distance to it) is shown. The code is available on request.

Algorithm 1. Pseudocode of the detection module

```
imageBlob = cv2.dnn.blobFromImage(
 cv2.resize(frame, (300, 300)), 1.0, (300, 300),
 (104.0, 177.0, 123.0), swapRB=False, crop=False)
detector.setInput(imageBlob)
detections = detector.forward()
faces = self.get_list_face(detections, [w, h, w, h])
LPaint: List[FacePaintItemDto] = []
self.faceCommand = FaceCommand(
False, "unknown", 0.99, "unknown")
for _, box in faces.items():
        color = (7, 193, 255)
        fontcolor = "#ffc107"
        (startX, startY, endX, endY) = CoordinateDto.modify(box)
    face = frame[startY:endY, startX:endX]
    (fH, fW) = face.shape[:2]
    self.last_coordinates = CoordinateDto(
    fW, fH, startX, startY, endX, endY)
    if fW < 20 or fH < 20:
    continue
    # NOTE: Дистанция до лица
    Dconst = (220 * 100) / 20
    distance = (20 * Dconst) / (fW == 0 and 1 or fW)
        distance_text = "{:.2f} метров".format(
                    distance / 100)
```

The detection module's output is illustrated in Fig. 5.

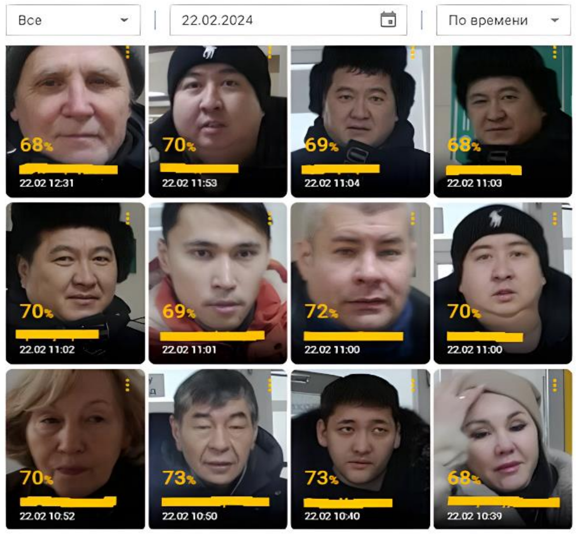


Fig. 5. Result of executing "detection module".

- Head Pose Estimation Module: Determining the vertical and horizontal position of the head based on the region of interest. The module is necessary for identifying the intention to enter, i.e., a direct gaze at the camera with a 10% tolerance. It determines the head position in 3D space, as shown in Algorithm 2.

Algorithm 2. Pseudocode for the head position detection module.

```
  def get_head_pose_estimation (self, dto: ImageProcessingFpeDto):
        """
         Attributes
        img:numpy array
        Resturns
        image
        """
        blink = False
        size = dto.frame.shape
  gray=cv2.cvtColor(dto.frame,cv2.COLOR_BGR2GRAY)
        shape = None
        try:
  rects = dto.detector(gray, 0)
  for rect in rects:
  shape = dto.predictor(gray, rect)
  shape = np.array(face_utils.shape_to_np(shape))
        except:
            return (dto.frame, -99, -99, -99, False)
```

```
        if shape is not None:
            image_points = np.array([
                (shape[33, :]),
                (shape[8, :]),
                (shape[36, :]),
                (shape[45, :]),
                (shape[48, :]),
                (shape[54, :])
            ], dtype="double")
            focal_length = size[1]
            center = (size[1]/2, size[0]/2)
            camera_matrix = np.array(
                [[focal_length, 0, center[0]],
                    [0, focal_length, center[1]],
                    [0, 0, 1]], dtype="double")
            dist_coeffs = np.zeros((4, 1))
            (_, rotation_vector, translation_vector) =
cv2.solvePnP(self.model_points,
    image_points,
    camera_matrix,
    dist_coeffs,
    flags=cv2.SOLVEPNP_ITERATIVE)
    rvec_matrix = cv2.Rodrigues(rotation_vector)[0]
    proj_matrix = np.hstack((rvec_matrix,
translation_vector))
    eulerAngles =
cv2.decomposeProjectionMatrix(proj_matrix)[6]
    pitch, yaw, roll = [math.radians(_) for _ in
eulerAngles]
    pitch = math.degrees(math.asin(math.sin(pitch)))
    roll = -math.degrees(math.asin(math.sin(roll)))
    yaw = math.degrees(math.asin(math.sin(yaw)))
    if dto.draw_cube == True:
    dto.frame = self.draw_annotation_box(
    dto.frame,
    points = image_points,
    rotation_vector = rotation_vector,
    translation_vector = translation_vector,
    camera_matrix = camera_matrix,
    dist_coeffs = dist_coeffs,
    color= (128, 255, 128))
    return (dto.frame, pitch, roll, yaw, blink)
    else:
return (dto.frame, -99, -99, -99, False)
```

- User Identification Module: Matches detected faces with those stored in the database.
- ACMS Module: Integrates with the Perco-S-20 system to manage and control access.

Algorithm 3 shows a fragment of the FaceSignHandler file (which opens the passage for the specified turnstile controller).

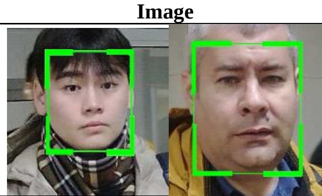
Algorithm 3. Pseudocode for the ACS module

```
    class FaceSignCommand():
        def __init__(self, index: int, controller):
    self.index= index
    self.controller = controller
    @mediatr.handler
    class FaceSignCommandHandler():
        def handle(self, request: FaceSignCommand):
            if request.controller is not None:
    answer= r.post("%sData/ExecuteData" %
(settings_provider.settings.percoservice),
    data={
    'commandpath':request.controller.commandpath,
    'devicename': request.controller.devicename,
    'typename': request.controller.typename
    }
    )
    if str(answer.status_code) != "200":
    print(
    bcolors.FAIL + str(" * [ERROR] %s" %
(answer.reason)) + bcolors.ENDC)
    return answer.status_code
```

- Training Module: Continuously improves the model by learning from new data.

Head position detection module. Determination of the vertical and horizontal position of the head in the area of interest. The module is needed to determine the login intent, i.e., a direct look at the camera with 10% tolerances determines the position of the head in 3-dimensional space (Table III).

TABLE III. FACTORS AND THEIR PERMISSIBLE VALUES

№	State	Image
1	Conditions met, identified	
2	Conditions met, not identified	
3	The head tilt angle along the axes is not acceptable, the identification model will not be used	

The region of interest is transformed into a format suitable for the input of the trained model (vector), and the model outputs a possible person and a match percentage. The match percentage should not be less than 70%. It is crucial that the identification process is sufficient for real-time operation, meaning the time taken to process a single frame should be no longer than the time between frame changes. For example, if the camera provides a video stream at 8 frames per second, the maximum time for image processing should be ~125 ms. The higher the frame rate, the less time is available for processing each frame; otherwise, the recognition process starts lagging and "working in the past".

To address this problem, we moved image (matrix) calculations from the central processor to the graphics processor using CUDA technology. We used CUDA technology to perform and parallelize calculations on the server's video adapter (RTX 2070). In implementing the system, we did not develop our own parallel computing methods but used the capabilities offered by CUDA and libraries that work with this technology (OpenCV, Dlib). As a result, the recognition speed was accelerated by an order of magnitude: CPU: 1–2 fps vs. GPU: 9–10 fps.

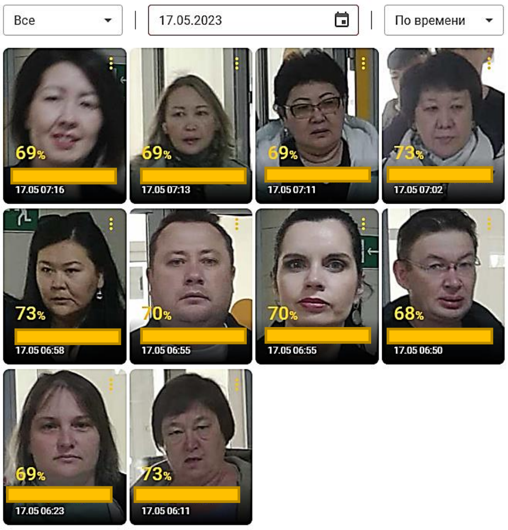
In addition to the time spent on processing within the system, the response time from external systems, such as the ACMS turnstile system, must also be considered. According to our calculations, the operations of opening the turnstile, recording passage data, and receiving a response take an average of 2–4 s. Based on the above, we assume that the adequate time for the entire process from the camera to recording the passage in SPortal should be no more than 2 s (Fig. 1).

The ACMS module is implemented as an integration with the university's access control system API. It includes the following operations: getting the turnstile status and opening/closing the passage.

The training module organizes work with the image database for training and fine-tuning the recognition model. Currently, the database contains 25,008 "raw" images (413 employees and students, as the system is in the testing phase, and not all employees and students can use FaceID due to server capacity limitations). Only "new" images are used for fine-tuning the recognition model, as "old" ones are marked as previously trained in a separate table. Fine-tuning is only performed for those employees whose average recognition threshold is below 70% (the recognition percentage can be seen in Fig. 6). As a rule, adding new images positively impacts recognition accuracy.

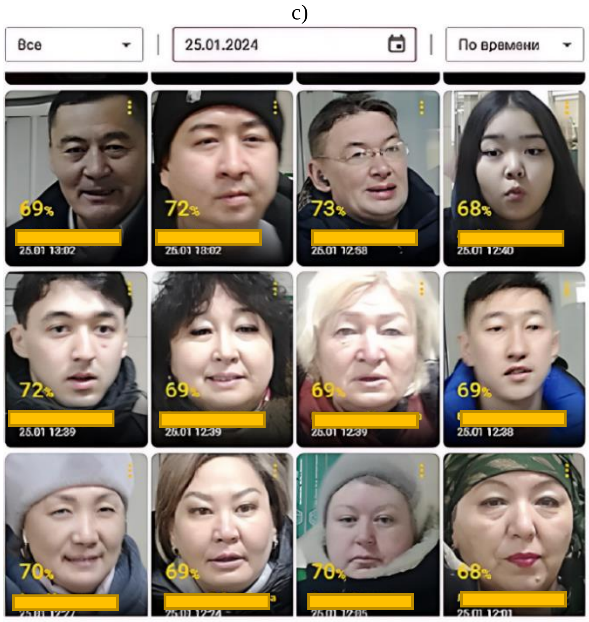
Let's consider the change in recognition quality using one employee as an example. It is important to understand that during recognition, we do not seek the highest possible percentage but check the acceptable threshold of 70%. If the percentage is consistently below 70%, fine-tuning is performed. The system can also choose the recognized face with the minimum threshold, but fine-tuning affects the frequency of achieving the minimum threshold. Several new images from the camera installed and integrated with the ACMS were added to the initial images.

a)



b)

c)



d)

Fig. 6. a) initial set of images in the database (64 photographs); b) initial recognition result 68%; c) additional training of the system - adding several photos to the database; d) recognition result after additional training 73%

An example of positive fine-tuning is shown in Fig. 6, where the recognition percentage increased by ~4–5%.

By analyzing the data on passages, we can observe how the recognition rate changes depending on different times of the day (morning, afternoon, and evening), which corresponds to the operational schedule of the educational institution. Passage data for employees was collected from 7:00 AM to 9:00 PM, as educational institutions do not operate at night. The system can be used for institutions with nighttime operations, provided there is additional lighting. The total number of passages for one employee over a period of 6 months was 254.

Fig. 7 shows how the recognition percentage changed as a result of system fine-tuning for different times of the day.

It is evident that under daytime lighting, even before fine-tuning, the recognition percentage was slightly higher than with additional artificial lighting (morning and evening). Fine-tuning the system after adding 12 images from the camera integrated with the ACMS improved the recognition percentage regardless of the time of day by almost 30%.

To integrate the access control system with the existing information system of the university, a web application "Employee Passes Through the Access Control System" was developed, which allows you to visualize for each employee his passes using a card or biometric identification, the date, time and place of the pass, as shown in Fig. 8.

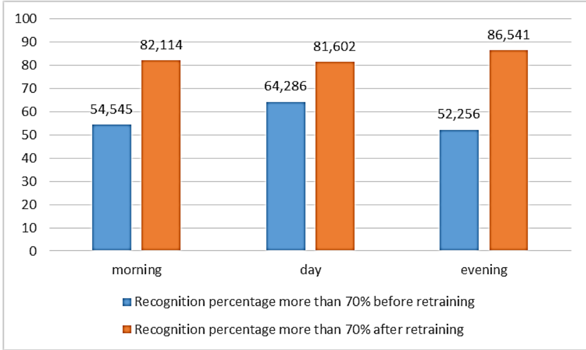


Fig. 7. Diagram of improvement in recognition percentage after additional training of the system.

At the moment of passing through the turnstile, the system analyzes the access rights of each employee and makes a decision to allow or deny passage. In this case, each event is registered in the system database indicating the time of passage. The web application also displays how the employee has been identified using Face ID  or a card .

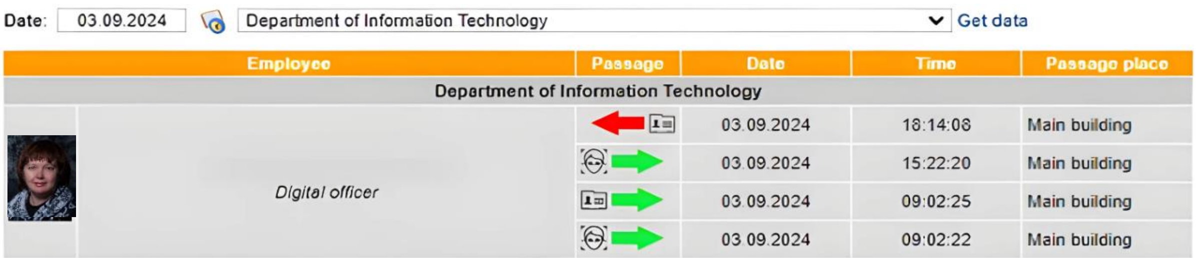


Fig. 8. Example of displaying data on an employee's passage.

The electronic authentication system we have developed is aimed at improving the overall security of the organization and convenience for employees and students.

To evaluate the developed system, several similar solutions are analyzed in Table IV, which examines the methods, limitations, and advantages of each.

TABLE IV. COMPARISON BETWEEN EXISTING AND PROPOSED APPROACH

Link	Method	Limitations	Advantages	Result
[43]	PCA–LGBPHS PCA–GABOR Wavelets	Illumination condition	Complexity	95%
[44]	GW-LDA	High processing time	Illumination invariant and reduce the dimensionality	88.12%
[45]	CNN–LSTM–ELM	High processing time	Automatically learn feature representations	90.60%
Proposed Approach	DNN	System response time from 70 ms to 125 ms	Lighting invariance, various poses and facial expressions, the ability to further train the system	93.54%

V. CONCLUSION

Methods have been developed to improve the quality of face recognition by retraining the model on new data, which made it possible to adapt an existing model to the specific conditions and requirements of the university, improving its ability to distinguish faces in real scenarios.

A software package with a recognition module for institutional security systems integrated with access control systems has been developed. The database of primary images was supplemented with images from cameras of the "Safe University" system to increase the recognition percentage; as a result of additional training of the system, it increased by ~4–5%.

Various lighting conditions and shooting angles were also considered, which influenced the increase in the percentage of identification satisfying the condition of more than 70% and the number of such identifications increased by more than 17%.

The trained model was optimized for fast and accurate operation on server hardware, which provided recognition speeds from 70 ms to 125 ms. After completion of training and additional training, the developed system was tested on more than 1000 university employees and visitors.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

N. Denissova conducted the research, analyzed the data, and wrote the article. I. Dyomina, A. Tlebaldinova and K. Apayev collected and pre-processed the data. N. Denissova and I. Dyomina analyzed and improved proposed an approach and reviewed the document. All authors approved the final version.

FUNDING

The work was carried out with the support of grant funding for scientific and (or) scientific and technical projects for 2023–2025 from the Ministry of Science and Higher Education of the Republic of Kazakhstan (grant No. AP19677501 "Development of intelligent services for a smart campus with integration into the university information ecosystem").

REFERENCES

[1] Y. Kortli, M. Jridi, A. A. Falou, and M. Atri, "Face recognition systems: A survey," *Sensors*, vol. 20, no. 2, 342, 2020. doi: 10.3390/s20020342

[2] M. Andrejevic and N. Selwyn, "Facial recognition technology in schools: Critical questions and concerns," *Learning, Media and Technology*, vol. 45, no. 2, pp. 115–128, 2019. doi: 10.1080/17439884.2020.1686014

[3] Y. Tao, Y. He, and W. Zhang, "An application of face recognition technology in university classroom teaching," in *Proc. 2020 IEEE 2nd International Conference on Computer Science and Educational Informatization (CSEI)*, Xinxiang, China, 2020, pp. 190–193. doi: 10.1109/CSEI50228.2020.9142538

[4] R. Rameswari, S. N. Kumar, M. A. Aananth, and C. Deepak, "Automated access control system using face recognition," *Materials Today: Proceedings*, vol. 45, no. 2, pp. 1251–1256, 2021. doi: 10.1016/j.matpr.2020.04.664

[5] P. B. Jha, A. Basnet, B. Pokhrel, B. Pokhrel, G. K. Thakur, and S. Chhetri, "An automated attendance system using facial detection and recognition technology," *Apex Journal of Business and Management*, vol. 1, no. 1, pp. 103–120, 2023. doi: 10.61274/apxc.2023.v01i01.008

[6] Y. Ghorpade, H. Thakare, S. Sonawane, A. Dedhia, and S. Mukherjee, "Face recognition attendance monitoring systems," *International Journal of Advanced Research in Science*, vol. 4, no. 3, pp. 543–546, May 2024. doi: 10.48175/IJARSCT-18281

[7] M. H. Abdulabas and N. D. Al-Shakarchy, "Person identification based on facial biometrics in different lighting conditions," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 2, pp. 2086–2092, April 2023. doi: 10.11591/ijece.v13i2.pp2086-2092

[8] H. Wang, "Studies advanced in face recognition," in *Proc. the 3rd International Conference on Signal Processing and Machine Learning, Applied and Computational Engineering Applied and Computational Engineering*, vol. 6, no. 1, 2023, pp. 867–873. doi: 10.54254/2755-2721/6/20230946

[9] H. K. D. Menon and V. Janardhan, "Machine learning approaches in education," *Materials Today: Proceedings*, vol. 43, pp. 3470–3480, 2021. doi: 10.1016/j.matpr.2020.09.566

[10] J. Yan, X. Zhang, Z. Lei, and S. Z. Li, "Face detection by structural models," *Image and Vision Computing*, vol. 32, pp. 790–799, 2014. https://doi:10.1016/j.imavis.2013.12.004

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, pp. 91–99, 2015. doi: 10.48550/arXiv.1506.01497

[12] J. Li, Y. Wang, X. Zhang, W. Deng, and H. Shen, "Metric classification network in actual face recognition scene," *Computer Vision and Pattern Recognition*, vol. 1, 2019. doi: 10.48550/arXiv.1910.11563

[13] X. Wang, "Deep learning in object recognition, detection, and segmentation," *Foundations and Trends® in Signal Processing*, vol. 8: no. 4, pp. 217–382, 2016. doi: 10.1561/2000000071

[14] R. anjan, V. M. Patel, and R. Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 121–135, 2017. doi: 10.1109/ TPAMI.2017.2781233

[15] R. Hiramatsu, J. Kofuji, S. Kofuji, and L. Mattes, "FaceID: benchmarks for face detection algorithms using cell processor," *Technical Poster*, 2008.

[16] D. J. Robertson, R. Jenkins, and A. M. Burton, "Face detection dissociates from face identification," *Visual Cognition*, vol. 25, no. 7–8, pp. 740–748, 2017. doi: 10.1080/13506285.2017.1327465

[17] A. Azulay and Y. Weiss, "Why do deep convolutional networks generalize so poorly to small image transformations?" *Journal of Machine Learning Research*, vol. 20, no. 184, pp. 1–25, 2019.

[18] R. Yadav and P. Kacker, "Efficient technique for real-time face detection," *Materials Today: Proceedings*, vol. 56, pp. 3306–3311, 2022. doi: 10.1016/j.matpr.2021.10.118

[19] S. Sarraf and M. Kabia, "Optimal topology of vision transformer for real-time video action recognition in an end-to-end cloud solution," *Mach. Learn. Knowl. Extr.*, vol. 5, no. 4, pp. 1320–1339, 2023. doi: 10.3390/make5040067

[20] B. Wu, C. Xu, X. Dai, A. Wan, P. Zhang, Z. Yan, and P. Vajda, "Visual transformers: Token-based image representation and processing for computer vision," arXiv preprint, arXiv: 2006.03677, 2020.

[21] S. He, W. Chen, and K. Wang, "Region generation and assessment network for occluded person re-identification," *IEEE Transactions on Information Forensics and Security*, 2023.

[22] A. Serbetci and Y. S. Akgul, "Regularizing binary neural networks via ensembling for efficient person re-identification," *IEEE Access*, vol. 11, pp. 59446–59455, 2023.

[23] J. R. Babu, T. Balaji, and S. P. R. Borra, "Experimental investigations to face recognition using openCV for monitoring turnout time," *Journal of Theoretical and Applied Information Technology*, vol. 101. no. 11, June 2023.

[24] N. Kim, J.-H. Kim, and C. S. Won, "FAFD: Fast and accurate face detector," *Electronics*, vol. 11, no. 6, 875, 2022. doi: 10.3390/electronics11060875

[25] Suwarno and Kevin, "Analysis of face recognition algorithm: Dlib and OpenCV," *Journal of Informatics and Telecommunication Engineering*, vol. 4, no. 1, pp. 173–184, 2020. doi: 10.31289/jite.v4i1.3865

[26] A. Viswanathan1, E. Kuralamudhan, S. Navaneethan, and S. Veluchamy, "Smart attendance system using face recognition," *EAI Endorsed Transactions on Scalable Information Systems*, pp. 1–6, February 2024. doi: 10.4108/eetsis.5203

[27] M. G. Mohammed and A. I. Melhum, "Implementation of HOG feature extraction with tuned parameters for human face detection," *International Journal of Machine Learning and Computing*, vol. 10, no. 5, September 2020. doi: 10.18178/ijmlc.2020.10.5.987

[28] RR. H. P. Sejati, and R. Mardhiyyah, "Face landmark based face detection using OpenCV and Dlib," *Journal of Information Technology*, vol. 5, no. 2, pp. 144–148, 2021. doi: 10.36294/jurti.v5i2.2220 (in Indonesia)

[29] M. A. Iqbal, J. C. Yoon, M. U. S. Khan, and S. K. Kim, "Improved open world object detection using class-wise feature space learning," *IEEE Access*, vol. 4, 2016. doi: 10.1109/ACCESS.2017

[30] S. Maniar, S. Dhage, K. Sukhani, and K. Shah "Automated proctoring system using computer, vision techniques," in *Proc. International Conference on System, Computation, Automation and Networking (ICSCAN)*, 2021, pp. 1–6. doi: 10.1109/ICSCAN53069.2021.952641

[31] N. Prasad, B. Rajpal, K. K. R. Mangalore, R. Shastri, and N. Pradeep, "Frontal and non-frontal face detection using Deep Neural Networks (DNN)," *International Journal of Research in Industrial Engineering*, vol. 10, no. 1, pp. 9–21, 2021. doi: 10.22105/riej.2021.264744.1177

[32] Y. Bengio, I. Goodfellow, and A. Courville, *Deep Learning*, Massachusetts: MIT Press, 2017.

[33] K. C. Kirana, S. Wibawanto, N. Hidayah, G. P. Cahyono, and K. Asfani, "Improved neural network using integral-RELU based

prevention activation for face detection," in *Proc. International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, Denpasar, Indonesia, pp. 260–263, 2019. doi: 10.1109/ICEEIE47180.2019.8981443

[34] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," *Neural Information Processing Systems*, vol. 25, 2012. doi: 10.1145/3065386

[35] S. H. S. Basha, S. R. Dubey, V. Pulabaigari, and S. Mukherjee, "Impact of fully connected layers on performance of convolutional neural networks for image classification," *Neurocomputing*, vol. 378, pp. 112–119, 2020. doi: 10.1016/j.neucom.2019.10.008

[36] Y. Ho and S. Wookey, "The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling," *IEEE Access*, vol. 8, pp. 4806–4813, 2020. doi: 10.1109/ACCESS.2019.2962617

[37] Open Source Computer Vision (OpenCV). [Online]. Available: https://docs.opencv.org/4.x/d5/d1f/calib3d_solvePnP.html

[38] S. E. Ivanov, N. N. Gorlushkina, and A. Govorov, "The recognition and classification of objects based on the modified distance metric," *Procedia Computer Science*, vol. 136, no. 7, pp. 210–217, 2018. doi: 10.1016/j.procs.2018.08.257

[39] M. I. Wahyudi, E. W. Wibowo, and Sopiullah, "Web-based face recognition using line edge detection and euclidean distance method," *Edumatic: Jurnal Pendidikan Informatika*, vol. 6, no. 1, pp. 135–142, 2022. doi: 10.29408/edumatic.v6i1.5525

[40] B. Escoffier, O. Spanjaard, and M. Tydrichová, "Algorithmic recognition of 2-euclidean preferences," in *Proc. 2023 the 26th European Conference on Artificial Intelligence*, 2023, pp. 637–644. doi: 10.3233/FAIA230326

[41] A. Tlebaldinova, N. Denissova, and D. Kassymkhanova, "Application of a scenario approach in development of a recognition system of vehicle identification numbers," in *Proc. International Conference on Modeling Simulation and Applied Optimization (ICMSAO)*, Istanbul, Turkey, 2015, pp. 1–4. doi: 10.1109/ICMSAO.2015.7152255

[42] D. Muratuly, N. F. Denissova, and I. V. Krak, "Information technology for a proctor to detect violations during the exam," *Cybernetics and Systems Analysis*, vol. 58, no. 6, pp. 983–990, 2022. doi: 10.1007/s10559-023-00533-x

[43] H. Cho, R. Roberts, B. Jung, O. Choi, and S. Moon, "An efficient hybrid face recognition algorithm using PCA and GABOR wavelets," *International Journal of Advanced Robotic Systems*, vol. 11, no. 4, 2014. doi: 10.5772/58473

[44] A. A. Fathima, S. Ajitha, V. Vaidehi, M. Hemalatha, R. Karthigaiveni, and R. Kumar, "Hybrid approach for face recognition combining Gabor wavelet and linear discriminant analysis," in *Proc. 2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS), Bhubaneswar*, India, 2015, pp. 220–225. doi: 10.1109/CGVIS.2015.7449925

[45] J. Sun, Y. Fu, S. Li, J. He, C. Xu, and L. Tan, "Sequential human activity recognition based on deep convolutional network and extreme learning machine using wearable sensors," *Journal of Sensors*, 8580959, 2018. doi: 10.1155/2018/8580959