

Solving specific tasks of face image processing and analysis using artificial intelligence based tools

Valerii Sokurenko^{1,†}, Bohdan Rusyn^{2,†}, Oleksandr Manzhai^{1,†}, Vitalii Nosov^{1,†}, Svitlana Luchyk^{1,*,†} and Vasil Luchyk^{1,†}

¹ Kharkiv National University of Internal Affairs, L. Landau Avenue 27 61080 Kharkiv, Ukraine

² Karpenko Physico-Mechanical Institute of the NAS of Ukraine, Naukova Street 5 79601 Lviv, Ukraine

Abstract

The negative demographic and socio-economic consequences of war have led to an intensification of criminal trends in Ukraine. Therefore, there is a constant need to accelerate the processes of integrating artificial intelligence technologies into modern criminal analysis to enhance the effectiveness of law enforcement agencies. This article examines the resolution of specific tasks in facial image processing and analysis using artificial intelligence-based tools. The authors conducted a comparative analysis of software products that provide facial recognition and localization services in images. They completed an in-depth study of facial recognition models, specifying their advantages and disadvantages, in order to utilize them most effectively in combination. A Python script was developed that implements an image processing pipeline which prioritizes the use of the Dlib detector and then, in case of its failure, switches to a backup MTCNN detector. The proposed hybrid approach aims to optimize both detection accuracy and efficiency, which is confirmed by successful processing of a wide range of images. In the article, the authors emphasize the presence of several limitations in applying this framework. Therefore, there is a need for research to be continued, particularly in the direction of optimizing the scaling coefficient and integrating additional quality metrics for automatic evaluation of cropped faces.

Keywords

facial recognition technologies, criminal analysis, CNN facial detector models, hybrid approach, two-stage pipeline, efficiency

1. Introduction

War, economic hardships, and mass migration have significantly affected the population size in Ukraine. According to IMF estimates, Ukraine's population in 2025 amounts to 32.9 million people. Compared to the pre-war year of 2021, this reduction constituted 21.9% [1]. The negative demographic and socio-economic consequences of war have led to an intensification of criminal trends in Ukraine. According to data from the Prosecutor General's Office, over twelve months of 2022, the number of crimes committed in Ukraine exceeded the number of crimes committed in 2021 (321,443 crimes) and the number of crimes committed in 2020 (360,662 crimes). Over the 12 months of 2024, law enforcement agencies registered 492,479 crimes with corresponding criminal proceedings, of which 194,688 criminal proceedings involved charges against specific individuals. Over 6 months of 2025, law enforcement agencies have already registered 327,847 crimes with corresponding criminal proceedings, of which 101,399 criminal proceedings involved charges against specific individuals [2].

Due to the shortage of human resources, law enforcement agencies are compelled to maximize the use of cutting-edge information technologies and artificial intelligence (AI) for processing and

*AISSE-2025: International Workshop on Applied Intelligent Security Systems in Law Enforcement, October, 30–31, 2025, Vinnytsia, Ukraine

^{1*} Corresponding author.

[†] These authors contributed equally.

✉ rector_hnuvs@ukr.net (V. Sokurenko); b.rusyn.prof@gmail.com (B. Rusyn); sofist@ukr.net (O. Manzhai); vitnos.g@gmail.com (V. Nosov); luchiksvitlana@gmail.com (S. Luchyk); luchik-vasil@ukr.net (V. Luchyk)

ORCID 0000-0001-8923-5639 (V. Sokurenko); 0000-0001-8654-2270 (B. Rusyn); 0000-0001-5435-5921 (O. Manzhai); 0000-0002-7848-6448 (V. Nosov); 0000-0003-0757-1140 (S. Luchyk); 0000-0001-6007-910X (V. Luchyk)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

analyzing large volumes of materials. For example, the Palantir system is used to review data arrays and conduct searches and processing of large volumes of information according to specific parameters already at the stage of pre-trial investigation. The Microsoft Azure program has enabled Ukrainian prosecutors and investigators to investigate war crimes committed by Russian military personnel. Clearview AI technology is actively employed for verifying individuals at checkpoints, identifying deceased soldiers and prisoners of war, and searching for missing persons. Facial recognition technology has become one of the most powerful applications of artificial intelligence for law enforcement agencies. The task of extracting and normalizing facial images, bringing a large number of heterogeneous images (photographs containing faces of specific individuals who were photographed both directly and indirectly through photographic documentation of objects) to a unified format is critically important in solving operational and service tasks in the IT units of the National Police of Ukraine. Therefore, it is important for law enforcement officers to deepen their understanding of all capabilities of this technology to prevent and investigate criminal acts against society, and to make criminals' use of cutting-edge digital technologies more complicated.

The goal of this work is to improve the effectiveness of law enforcement agencies by developing and experimentally testing a hybrid approach to processing and recognizing facial images based on artificial intelligence tools.

The main task of the research is to determine the optimal combination of face detection models that improves the accuracy, speed, and reliability of identifying individuals in digital images, particularly in conditions of low quality, varying lighting, or shooting angles.

2. Related works

Over recent years, facial recognition technologies have undergone significant changes due to the implementation of deep learning methods for neural networks. The main application areas of these technologies include biometric authentication, video surveillance, forensic investigations, digital identification, and others [3-7]. Publication [7] presents a systematic review of scientific publications on current facial recognition algorithms. Deep learning of neural networks in facial recognition tasks involves the construction and training of multi-level models capable of automatically extracting characteristic facial features from raw pixel images. Common architectures of deep convolutional neural networks (CNN) in facial recognition tasks include: VGGNet [8], ResNet [9], InceptionNet [10] used as base frameworks; FaceNet [11] generates vector representations of faces (embeddings); ArcFace [12], CosFace [13], SphereFace [14] introduce modifications to the loss function to enhance discriminativeness. Before CNN training happens, the following procedures are performed: face detection with extraction of the facial region (for example, using the MTCNN [15] and RetinaFace [16] methods); alignment with normalization of the position of eyes, nose, and mouth; scaling and normalization of images; data augmentation, or the process of artificially generating new data from existing data, which includes rotation, mirroring, and adding noise to images to improve generalization. Such data augmentation allows for artificial expansion of the dataset by introducing minor modifications to the original data.

The CNN training process occurs on a large dataset with millions of images, where the CNN automatically learns to extract features instead of manual descriptor design. This process utilizes a loss function that quantitatively measures how “poorly” the model performs its task or how much its predictions differ from the true values. The training objective consists precisely in minimizing this loss function. At the final stage, the CNN transforms the facial image into a fixed-length vector (for example, 128 or 512 elements) that preserves semantic similarity (e.g., faces of the same person have similar vectors) and can be used for classification, search, and verification. The CNN is also validated on a test dataset that was not used during training.

At the same time, CNN facial recognition technologies encounter a number of various problems and challenges when implemented. These are highlighted by several researchers in their publications. Among the main problems identified is image variability (i.e. intra-class variation)

[17] that arises due to changes in lighting, viewing angle, facial expressions, accessories (such as glasses or masks), as well as image quality. Such factors significantly complicate person identification, reducing the accuracy of even modern deep learning models.

Another problem is model transfer to new domains (i.e. domain adaptation). It occurs when models trained on one type of data often demonstrate reduced effectiveness when applied in new conditions (for example, changes in cameras, environment, culture, or usage context). This limits the scalability and universality of recognition systems without prior fine-tuning [18].

In facial recognition, particularly in applied tasks, the problem of limited training data volume frequently arises. Many open datasets have imbalances in the representation of racial, gender, and age groups, leading to algorithmic bias [19]. Additionally, deep learning models may demonstrate higher accuracy for certain demographic groups compared to others, which is defined as model bias. This model property can have critical consequences in the context of ensuring human rights. For example, it is known that some models have significantly higher error rates when recognizing individuals with dark skin color or atypical facial features [20].

Finally, the unregulated deployment of facial recognition technologies raises significant societal concerns regarding privacy rights and informed consent [22-23]. Consequently, the various challenges and limitations encountered by practitioners, particularly law enforcement personnel, in facial recognition applications require comprehensive analysis of different CNN facial detector models to identify their respective advantages and limitations..

3. Materials and methods

Pattern recognition systems typically operate in training and testing modes. During the training phase, the feature space is partitioned into recognition classes for the purpose of constructing decision rules. An important section of pattern recognition theory is automatic classification or cluster analysis of input data [24]. A cluster consists of a set of similar (analogous) pattern realizations that can be separated from other objects according to specific criteria. Since cluster analysis lacks an array of class identifiers for the realizations, this process is also termed "unsupervised learning." The clustering process involves not only the search for similar realizations but also the formation of decision rules for each cluster. A priori information about the number of clusters and their distribution in the recognition feature space significantly simplifies this process. The main approaches, distinguished by their field of knowledge and scientific direction for solving pattern recognition problems, are:

1. Algebraic approach, whose main advantage is simple decision rules. The primary disadvantage of this approach lies in low recognition reliability, as it does not account for uncontrolled factors that influence the recognition process;
2. Geometric approach, characterized by universality, clarity, and simplicity of recognition algorithm interpretation;
3. Statistical approach, which employs statistical characteristics for data analysis;
4. Biological approach, which includes artificial neural networks. Algorithms within this approach model cognitive processes occurring in human brain nerve cells. The main disadvantage of the biological approach is high sensitivity to the dimensionality of the recognition feature space;
5. Network approach (semantic networks, frames, Petri nets, decision trees, etc.). The advantages of this approach include model simplicity, possibility for extension and complexity enhancement, while the main disadvantage is the complexity of constructing decision rules;
6. Fuzzy approach, developed based on the algebraic approach and serving as a competitor to the statistical approach. This approach allows modeling of pattern recognition processes that a priori overlap in the recognition feature space. However, it is not adapted for optimizing the parameters of recognition system functionality;

7. Game-theoretic approach, whose decision rules are characterized by high complexity and low recognition reliability.

Since all main approaches, except the algebraic one, intersect with the geometric approach, the formation of a general decision-making theory is most justified within the framework of the geometric approach. Within the geometric approach, pattern recognition theory is based on two fundamental principles:

1. The maximum-distance principle, whereby decision rules are constructed by maximizing the average inter-class distance.
2. The minimum-distance principle, whereby decision rules are constructed under the condition of minimizing the average distance of pattern realization to its class center.

Implementation of these principles constitutes a necessary condition for achieving maximum recognition reliability, which is determined by the total probability of correct decision-making:

$$P_t = p_1 D_1 = p_2 D_2 \quad (1)$$

where p_1, p_2 are unconditional probabilities, D_1, D_2 are the first and second reliabilities, respectively.

Suppose it is necessary to find pattern N , which is described by $N_i, i = 1...n$, features, each of which possesses $m_j, j = 1...m$ properties. Thus, the pattern can be described by a matrix of dimension $m \times n$:

$$N = (N_{11} \ \cdots \ N_{1m} \ \vdots \ \ddots \ \vdots \ N_{n1} \ \cdots \ N_{nm}) \quad (2)$$

Suppose it is necessary to identify within an image array the specific image that corresponds to pattern N . To accomplish this, we apply a known function f to pattern N :

$$f(N) = (f(N_{11}) \ \cdots \ f(N_{1m}) \ \vdots \ \ddots \ \vdots \ f(N_{n1}) \ \cdots \ f(N_{nm})) \quad (3)$$

Let the image array be denoted as $K = K_1, K_2, \dots, K_s$. Each image is described by a matrix

$$K^p = (K_{11}^p \ \cdots \ K_{1m}^p \ \vdots \ \ddots \ \vdots \ K_{n1}^p \ \cdots \ K_{nm}^p) \quad (4)$$

and the action of function f on matrix (3) is described accordingly by the functional matrix

$$f(K^p) = (f(K_{11}^p) \ \cdots \ f(K_{1m}^p) \ \vdots \ \ddots \ \vdots \ f(K_{n1}^p) \ \cdots \ f(K_{nm}^p)) \quad (5)$$

For the pattern to correspond to an image from the array, the following inequality must be satisfied:

$$\begin{aligned} |f(N) - f(K^p)| < \epsilon, \quad |f(N_{ij}) - f(K_{ij}^p)| < \epsilon^p, \\ i = \underline{1...n}, \quad j = \underline{1...m}, \quad p = \underline{1...s} \end{aligned} \quad (6)$$

The image can be considered found if the following condition is satisfied:

$$\sqrt{\sum_{p=0}^s (\epsilon^p)^2} \rightarrow \min. \quad (7)$$

In cases where it is necessary to reduce or increase the weight of pattern realizations that differ significantly from one another, the generalized power distance is applied:

$$d(f(N_{ij}), f(K_{ij}^p)) = \sqrt[r]{\sum_{p=0}^s (f(N_{ij}) - f(K_{ij}^p))^h}, \quad (8)$$

where r is the parameter responsible for progressive weighting of large distances between objects; h is the parameter responsible for gradual weighting of differences along individual coordinates.

In practice, the fuzzy compactness hypothesis of pattern realizations applies, as classes inherently overlap and exhibit indistinct boundaries. Consequently, the application of the aforementioned deterministic distance-based proximity criteria in classification tasks fails to achieve clear partitioning of the feature space into distinct recognition classes. To address this limitation in pattern recognition applications, the Mahalanobis distance has been adopted [25]:

$$d\left(f\left(N_{ij}\right), f\left(K_{ij}^p\right)\right)=\left\|f\left(N\right)-f\left(K^p\right)\right\|^T \cdot W^{-1}\left(f\left(N_{ij}\right), f\left(K_{ij}^p\right)\right), \quad (9)$$

where T denotes the transpose symbol for the column vector; W^{-1} represents the inverse covariance matrix.

During the analysis and synthesis of learning-capable recognition systems, an information measure in the form of (9) is widely employed as a general measure of pattern proximity (similarity).

$$C_l = 1 - E, \quad (10)$$

where E is the normalized information measure that represents the measure of recognition class diversity. In practical applications of information synthesis for learning-capable recognition systems, the Shannon entropy measure and the Kullback information measure have gained the most widespread adoption [26]. The normalized Shannon entropy criterion of functional efficiency has the form

$$E = \frac{H_0 - H(\gamma)}{H_0} \quad (11)$$

where H_0 is the unconditional average entropy:

$$H_0 = - \sum_{l=1}^M p(\gamma_l) \log_2 p(\gamma_l) \quad (12)$$

$H(\gamma)$ represents the a posteriori conditional entropy characterizing the residual uncertainty after decision-making:

$$H(\gamma) = - \sum_{l=1}^M p(\gamma_l) \sum_{m=1}^M p\left(\frac{\mu_m}{\gamma_l}\right) \log_2 p\left(\frac{\mu_m}{\gamma_l}\right) \quad (13)$$

In expressions (12) and (13), the following notations are adopted: p_l represents the unconditional (a priori) probability of accepting hypothesis l ; p_{ml} represents the a posteriori conditional probability of accepting hypothesis m given that hypothesis l was a priori accepted; M denotes the number of alternative hypotheses. In practical applications, the following assumptions are commonly made:

1. Decisions are binary in nature ($M = 2$).
2. Given that the recognition system operates under a priori uncertainty conditions, the assumption of equiprobable hypotheses is justified according to the Bernoulli-Laplace principle.

$$p(\gamma_1) = p(\gamma_2) = \dots = p(\gamma_m) = \frac{1}{M} \quad (14)$$

Then criterion (11), taking into account expressions (12)–(14), assumes the form:

$$E = 1 + \frac{1}{2} \sum_{l=1}^2 p(\gamma_l) \sum_{m=1}^2 p\left(\frac{\mu_m}{\gamma_l}\right) \log_2 p\left(\frac{\mu_m}{\gamma_l}\right) \quad (15)$$

In law enforcement systems for facial recognition, minimal error (maximum accuracy) and controlled error levels are required, which directly impacts mathematical modeling. Therefore, mathematical models are refined through probabilistic components, specialized loss functions, and multimodal architectures to ensure minimization of critical errors. The models must account for the following aspects:

1. Optimization of accuracy metrics (in modern machine learning and deep learning tasks, optimization algorithms play a critical role, as they determine the efficiency and speed of model training. Specifically, these algorithms aim to minimize the loss function by updating model parameters based on gradient information).
2. Bayesian interpretation and confidence levels (in such systems, it is important to obtain probability estimates of an individual's membership in a particular class). The model often incorporates a posteriori probabilities, that is, the probability of finding an image after the occurrence of a specific event.
3. Robustness to capture conditions (allowing facial recovery under low quality or partial occlusion). Several tasks related to robustness are distinguished. Robust stability ensuring system stability under all admissible deviations of the image object model from the nominal.
4. Multimodal models (combination of features from different biometric channels (voice, gait, 2D+3D data) allows for improved recognition accuracy).
5. Adaptive thresholds and calibration (discrimination thresholds are selected depending on the task context).

Therefore, in law enforcement activities, the mathematical modeling process of facial recognition shifts from purely classical accuracy optimization to controlled risk minimization. This means that explicit error cost components, a priori scenario probabilities, and calibrated a posteriori estimates are incorporated into the formalism. This stimulates the integration of multi-level representations and quality assessment mechanisms for individual data transformation processes, which, in turn, allows for adaptation of thresholds and re-verification algorithms before decision-making.

4. Results

We conducted a comparative analysis of software products that provide facial recognition and localization services in images. Information regarding the key features of these tools and their availability for free use is presented in Table 1.

Table 1

Comparative analysis of products that provide recognition and cropping services for objects in images

Service	Primary Specialization	Key Facial Cropping Capabilities	Features	Cost
<i>Imgix</i> (is.gd/f2BQz3)	Real-time image processing and optimization	Automatic cropping based on points of interest (including faces), scaling, formatting, filters, CDN	Dynamic image processing for websites and applications, caching	Free trial available. Paid plans: \$25/month from

<i>Crop.photo</i> (is.gd/MeG3j8)	Automated cropping for facial recognition systems	Automatic detection of multiple faces, selection of optimal regions	Optimized for data preparation for training AI facial recognition models	Free available. Paid plans: \$10/month	plan from
<i>Frame-A-Face</i> (is.gd/k4ROQy)	Intelligent image cropping	Face detection for object isolation, batch size uniformity	Batch image processing, support for specified cropping dimensions	Free available. Paid plans: \$30/month	trial from
<i>Cloudinary</i> (is.gd/WlqgXU)	Comprehensive solution for media management and delivery	API for automatic face cropping, multifunctional transformations, image optimization, CDN, asset management, video	Comprehensive platform for developers working with images and video	Free available. Paid plans: \$89/month	plan from
<i>Arya.ai</i> (is.gd/TrlGQk)	AI services and APIs for image processing	Intelligent detection and cropping of key visual regions	Focus on AI solutions, identification of focal zones for precise cropping	Free available. Pricing upon request	trial
<i>Sirv</i> (is.gd/gVr1zu)	Management and optimization of images and video in CDN	Automatic face detection using neural networks, cropping parameters, dynamic imaging, CDN, 360-degree rotation, video optimization	Cloud service for rapid media content delivery and transformation.	Free available. Paid plans: \$19/month	plan from
<i>Imagga</i> (is.gd/bVeofm)	AI services for image analysis and processing	Image composition analysis, identification of optimal cropping regions, image tagging, categorization, color analysis	Specializes in image understanding through AI, composition-aware cropping.	Free available. Paid plans: \$79/month	plan from
<i>ZOOP</i> (is.gd/lC30D9)	Identity verification, fraud	Advanced image processing technology	Oriented toward services	Free available. Pricing upon request	trial

	prevention	(detection, recognition, alignment), KYC, identity verification	requiring high-accuracy identity verification.	
--	------------	---	--	--

The findings in Table 1 highlight two critical constraints. First, effective processing of large-scale image datasets necessitates access to premium subscription tiers, which may be financially prohibitive for government agencies. Second, the majority of these platforms lack local deployment options, precluding their use for processing sensitive materials containing confidential information, particularly personal data, under current security protocols. To overcome these limitations, we propose a hybrid methodology incorporating dual face detection algorithms for automated facial detection, alignment, and cropping processes. We begin with a comprehensive examination of facial recognition models to optimize their combined implementation.

A typical facial recognition system pipeline consists of sequential stages, each performing a specific function. The efficiency of each stage is critical, as errors introduced in early phases can accumulate and significantly impact the final outcome.

4.1. Face detection

Face detection serves as the initial step, with the objective of localizing and isolating human faces within an image or video frame. This stage produces bounding boxes around each detected face. Various methods exist for face detection:

1. Traditional methods, which include:
 - The Viola-Jones algorithm, which employs Haar-like features and cascade classifiers.
 - Histogram of Oriented Gradients (HOG) combined with Support Vector Machine (SVM).
2. Deep learning-based methods: Convolutional Neural Networks (CNN), including Haar Cascades, Dlib HOG and Dlib CNN, Face Recognition (Dlib HOG + CNN), MediaPipe, MTCNN, RetinaFace, YOLOv5-Face, and OpenVINO Face Detection.

Analysis of recent research findings on these non-commercial face detection models [26-34], which employ deep learning architectures and are implemented as open-source libraries or frameworks in Python—including Haar Cascades, Dlib HOG, Dlib CNN, Face Recognition (Dlib HOG + CNN), MediaPipe, MTCNN, RetinaFace, YOLOv5-Face, and OpenVINO Face Detection—enabled their comparison across the following parameters:

F1-score is a metric used to evaluate the accuracy of classification models, particularly when classes in the dataset are imbalanced. This indicator is calculated as the harmonic mean (16) between precision (17) and recall (18).

$$F1\ score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}, \quad (16)$$

$$Precision = \frac{\text{Number of correctly detected faces } TP}{\text{All detected faces } TP + FP}, \quad (17)$$

$$Recall = \frac{\text{Number of correctly detected faces } TP}{\text{Total number of actual faces } TP + FN}, \quad (18)$$

where TP (True Positives) is a correctly detected faces; FP (False Positives) is a falsely detected faces; FN (False Negatives) is an undetected actual faces;

FPS (Frames Per Second) is a processing speed, that is, the number of images (frames) that can be processed per second. Latency is a processing latency per image.

The calculation results for the aforementioned parameters of face detection models employing deep learning architectures, as described in [27-35], are presented in Table 2.

Table 2

Quantitative parameters of face detection models employing deep learning architectures

Face Detection Model	Model Parameters							Processor
	FPS		Precision	Recall	F1-score	Latency (per image), ms		
	min	max				min	max	
Haar Cascades	25	30	0.70	0.60	0.65	30	40	CPU
Dlib HOG	5	10	0.85	0.75	0.80	100	150	CPU
Dlib CNN	10	15	0.98	0.94	0.96	250	400	CPU
Face Recognition	5	12	0.95	0.92	0.93	150	300	CPU
MTCNN	1	5	0.95	0.90	0.92	400	700	CPU
OpenVINO	80	120	0.95	0.91	0.93	5	10	CPU
Mediapipe	25	30	0.96	0.93	0.94	10	15	CPU
RetinaFace	15	25	0.98	0.96	0.97	50	120	GPU
YOLOv5-Face	20	45	0.97	0.94	0.95	10	20	GPU

Comprehensive analysis of the facial recognition model parameters defined in Table 2 allowed the following conclusions to be drawn.

Analysis of the models based on processing speed (FPS), specifically the frame rate range from minimum to maximum, enabled assessment of their suitability for real-time applications, essentially evaluating model performance. OpenVINO Face Detection emerges as the clear leader in this metric, achieving 80-120 FPS on CPU, making it an ideal candidate for surveillance systems and other high-throughput applications. Deep learning-based models such as MTCNN and Dlib HOG demonstrate the lowest speeds, with performance metrics of 1-5 and 5-10 FPS, respectively.

Comparison of models based on primary accuracy metrics (Figure 1a, 1b) revealed that Dlib CNN and RetinaFace demonstrate the highest accuracy performance with F1-scores exceeding 0.95. The data in Figures 1a and 1b indicate the models' capability to reliably detect faces with minimal false positive occurrences. The Haar Cascades model, despite being a classical approach, significantly underperforms across all accuracy metrics.

Examination of facial recognition models and comparison of their processing latency performance in milliseconds (Table 1) revealed that latency results correlate with FPS speed. The OpenVINO model exhibits the lowest latency (5-10 ms), confirming its high efficiency. At the opposite end of the spectrum is MTCNN with latency up to 700 ms, rendering it unsuitable for real-time applications but acceptable for offline analysis where accuracy takes priority.

To determine the trade-off between model accuracy (F1-score) and speed (maximum and minimum latency), we established a normalized accuracy-to-latency ratio by calculating

F1-score/Latency(ms,min)*100 and F1-score/Latency(ms,max)*100 metrics. The calculated data for each facial recognition model are displayed in Figure 2.

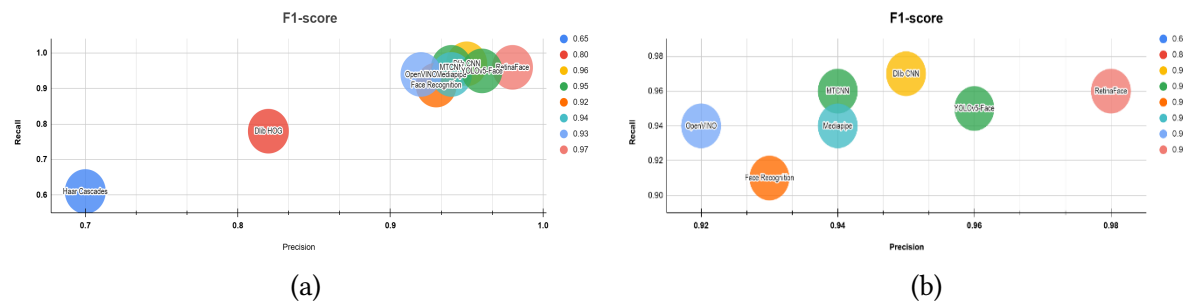


Figure 1: Facial recognition models by accuracy metric performance F1-score: (a) F1-score range 0.7-1.0; (b) F1-score range 0.92-0.98

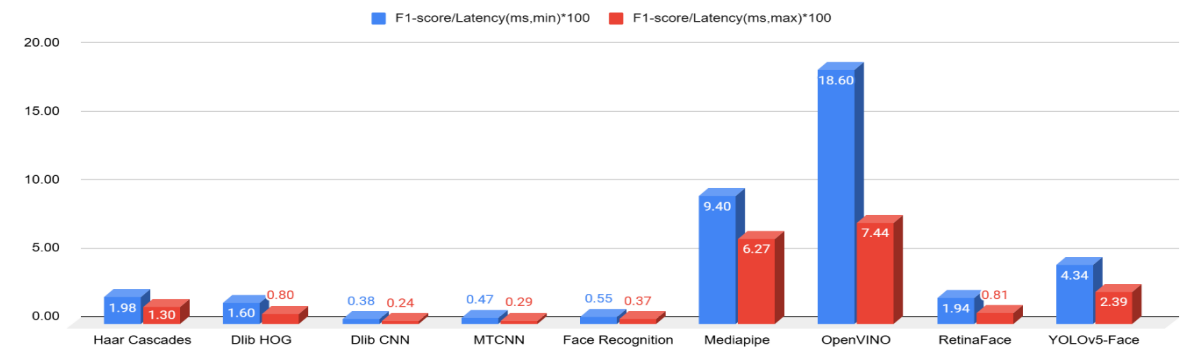


Figure 2: Comparative diagram of normalized accuracy-to-latency ratio metrics for different facial recognition models

As illustrated in Figure 2, the diagram reveals the fundamental challenge in face detection: balancing accuracy and speed. Models offering the optimal trade-off are OpenVINO and MediaPipe, which provide high accuracy (F1 > 0.92) with very low latency. Models with high accuracy but significant latency include Dlib CNN and MTCNN. RetinaFace represents the only GPU-based model demonstrating high accuracy with moderate latency. Haar Cascades exhibits low accuracy but relatively minimal latency.

Real-time facial recognition models require optimization for rapid image processing. This may include hardware acceleration such as GPU utilization, as well as quantization and pruning techniques to reduce model size and increase processing speed. Figure 3 demonstrates that GPU-accelerated models exhibit high performance while maintaining superior accuracy.

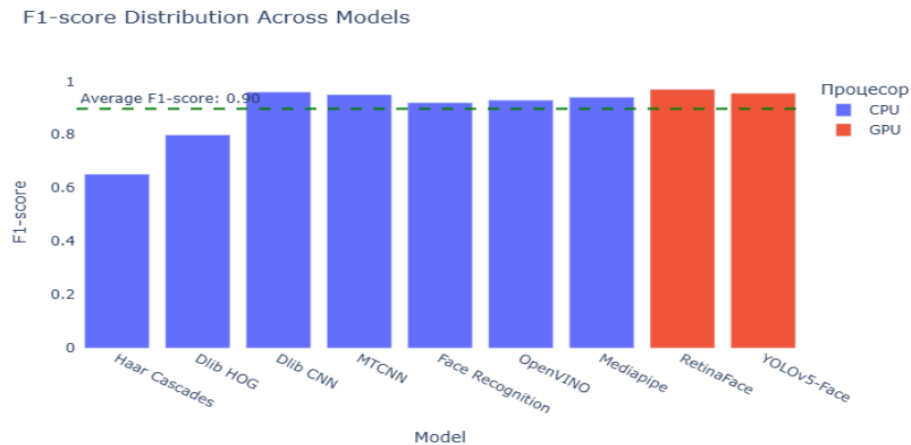


Figure 3: Distribution of facial recognition models by processor type and accuracy metric (F1-score)

4.2. Face alignment

Following face detection, the alignment stage aims to normalize the face to a standard position and orientation. This involves identifying facial landmarks such as eye corners, nose, and mouth corners, and transforming the image (e.g., rotation, scaling) so that these points are positioned consistently. Alignment reduces variations caused by pose and improves the consistency of features extracted in subsequent stages.

Reference [36] demonstrates that alignment enhances recognition accuracy by up to 6%. The techniques employed span from 2D affine transformations to more sophisticated 3D alignment methodologies. Within the DeepFace framework, alignment functionality is implemented by default, while the RetinaFace detector achieves superior alignment precision through its robust landmark detection capabilities. Preprocessing workflows incorporating facial alignment based on detector-identified landmarks have become established standard practice in contemporary facial recognition systems.

4.3. Face Representation and Feature Extraction

At this stage, the aligned facial image is transformed into a compact, discriminative numerical vector (embedding) that captures its essential characteristics. Two primary approaches exist for facial feature extraction:

1. Handcrafted features. These are based on manually designed algorithms for detecting edges, textures, shapes, or key points. Examples include LBP (Local Binary Patterns), HOG (Histogram of Oriented Gradients), and SIFT (Scale-Invariant Feature Transform). Advantages include interpretability and functionality with limited datasets, while disadvantages encompass the potential to miss the most discriminative information.
2. Learned features through deep learning. Convolutional Neural Networks (CNN) automatically learn hierarchical features from data. Initial layers extract edges and textures, while subsequent layers combine them into complex shapes. Examples include models such as DeepFace, FaceNet, VGG-Face, ArcFace, AdaFace, and MagFace. Advantages include high discriminative capability and robustness to variations. Disadvantages encompass requirements for large datasets, computational intensity, and reduced interpretability.

4.4. Face Matching and Classification (Verification/Identification)

This constitutes the final stage, which can be implemented in two distinct forms:

1. Verification (1:1). Comparison of two facial embeddings to determine whether they belong to the same individual. A similarity score is calculated (e.g., using cosine similarity or Euclidean distance) and compared against a threshold value.
2. Identification (1:N). Comparison of a query face embedding against a database of known face embeddings to find the closest match (or multiple matches).

Facial classification may employ Support Vector Machine (SVM), K-Nearest Neighbor (KNN) methods, or specialized similarity learning architectures such as Siamese Networks (SN), which are utilized for face matching through cosine similarity between output vectors.

5. Discussions

This investigation enabled us to comprehend the complexity, multifaceted nature, and critical importance of the entire facial recognition process, particularly within law enforcement applications. These technologies offer numerous advantages compared to other access control or

monitoring devices. However, our analytical findings revealed that the sequential nature of the facial recognition pipeline results in error accumulation. Enhancement of any individual stage can improve overall system accuracy. This is particularly relevant for the face detection stage. Reference [34] indicates that improving face detection accuracy can enhance overall recognition accuracy by up to 42%, while alignment contributes up to 6% improvement. Therefore, when selecting face detector models, we recommend considering the analytical results obtained in this study.

1. Model selection depends on task-specific requirements. For applications where maximum accuracy is paramount (e.g., biometric identification from photographs), RetinaFace or Dlib CNN models are recommended, utilizing graphics processing units (GPU) for acceleration when necessary. For real-time systems (e.g., surveillance, interactive applications), OpenVINO Face Detection or MediaPipe represent optimal choices due to their high processing speeds and low resource requirements.
2. Classical methods are outperformed by neural network approaches. The Haar Cascades algorithm significantly underperforms compared to modern models in both accuracy (F1-score ~ 0.65) and recall (~ 0.6). Despite relatively high processing speeds (up to 30 FPS), these models are unsuitable for systems where detection quality is critical.
3. Modern optimized models offer balanced solutions. The OpenVINO Face Detection model demonstrates exceptional CPU performance, achieving speeds up to 120 FPS while maintaining high accuracy (F1-score ~ 0.93). Similarly, the MediaPipe framework provides high accuracy (F1-score ~ 0.94) with low latency (10-15 ms), making these models optimal for diverse applications, including mobile applications and embedded systems.
4. A pronounced trade-off exists between accuracy and computational efficiency. Deep neural network-based models such as Dlib CNN and RetinaFace provide the highest detection accuracy (F1-score 0.95); however, their computational complexity results in significant latencies (50 to 400 ms per image). This constrains their use in real-time systems without specialized hardware acceleration.
5. CPU-based models demonstrate a broad performance spectrum. Among CPU-operating models, significant differentiation is observed. OpenVINO stands out as an extremely fast solution (80-120 FPS) with competitive accuracy (F1-score 0.93), making it ideal for embedded systems and Edge AI applications. Conversely, classical approaches like Haar Cascades, while fast, substantially underperform in accuracy (F1-score 0.65), limiting their applicability. Models such as MTCNN and Dlib CNN offer high accuracy but at the cost of substantial computational complexity and consequently high latency.
6. Hardware acceleration (GPU) impact is critical for high-performance systems. Models optimized for graphics processors (RetinaFace, YOLOv5-Face) demonstrate significantly better performance balance compared to CPU counterparts. They achieve high FPS rates (up to 45) while maintaining accuracy at F1-score ~ 0.95 levels. This makes them suitable for computer vision systems requiring real-time video stream processing.

Although the proposed framework may elicit academic discussion, such engagement potentially demonstrates that this research establishes novel pathways for subsequent studies.

We present a Python-based implementation utilizing artificial intelligence tools for automated facial detection, alignment, and cropping in images. The system employs a hybrid methodology incorporating two primary detection algorithms: Dlib's HOG (Histogram of Oriented Gradients) combined with SVM (Support Vector Machine) and MTCNN (Multi-task Cascaded Convolutional Networks). The framework accepts input images across multiple formats and quality specifications..

This automated facial cropping system operates through the integration and application of several fundamental scientific research contributions in computer vision and machine learning. Its

effectiveness and robustness are ensured through the utilization of advanced algorithms for face detection, landmark identification, image enhancement, and precise resampling (Figure 4).

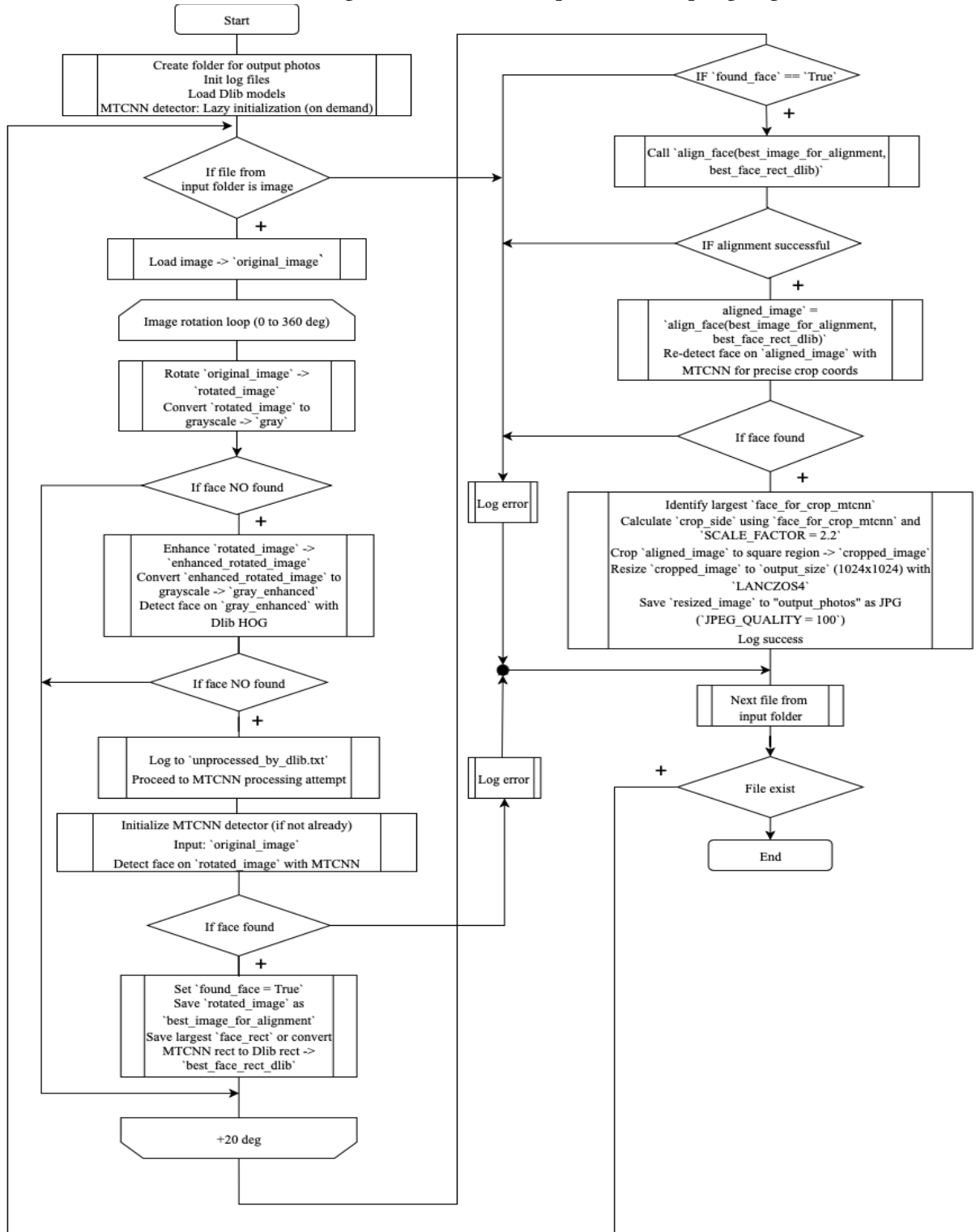


Figure 4: Program operation algorithm

The primary detector (Dlib HOG+SVM) employs a Histogram of Oriented Gradients (HOG) implementation combined with Support Vector Machine (SVM) from the Dlib library. This method efficiently extracts local gradient descriptors, which are input to a linear SVM classifier for binary classification—distinguishing faces from background. This methodology was comprehensively described in [37]. It should be noted that this approach is particularly effective for frontal and near-frontal faces.

To enhance the probability of face detection in images with varying orientations, the input image is iteratively rotated (in 20-degree increments across a 0-360 degree range) before applying the detector. In cases where faces are not detected on the original rotated image, preprocessing is applied to enhance contrast and brightness, followed by repeated detection attempts.

In the event of unsuccessful Dlib detector performance, the system transitions to MTCNN (Multi-task Cascaded Convolutional Networks) [38]. MTCNN is a convolutional neural network comprising three cascaded stages (P-Net, R-Net, O-Net), each performing tasks of facial region proposal generation, refinement, and facial landmark localization. This detector demonstrates high robustness to variations in pose, illumination, and facial scale. MTCNN initialization occurs dynamically, only when needed, to optimize system resource utilization. Face detection using MTCNN also incorporates iterative image rotations, analogous to the Dlib approach.

To enhance Dlib detection efficiency under challenging conditions, a function has been implemented that applies adaptive histogram equalization using CLAHE (Contrast Limited Adaptive Histogram Equalization) [39] in the LAB color space (on the luminance L channel) and additional linear brightness transformation in the HSV color space (on the V channel). CLAHE improves local contrast, enhancing facial visibility for the detector. It is important to note that these enhancements are applied only to image copies used for detection, while subsequent geometric transformations are performed on the original, unmodified rotated image to preserve maximum quality.

Following successful face detection, a Shape Predictor (the `shape_predictor_68_face_landmarks.dat` model from Dlib) is utilized to localize 68 facial landmarks. This model is based on the Supervised Descent Method (SDM), which was comprehensively described in [40]. The facial inclination angle is calculated based on the coordinates of the left and right eye centers. An affine transformation is applied to the image to align the face by positioning the eye line horizontally. This transformation employs Lanczos interpolation, which ensures high-quality pixel transformation. This resampling method, based on the application of the sinc function as a filter, is extensively discussed in digital image processing literature, particularly in [41]. Lanczos interpolation is recognized for its ability to minimize aliasing effects and preserve edge sharpness during image scaling, which is critically important for obtaining high-quality final photographs.

Following alignment, re-detection is performed on the aligned image to obtain precise coordinates. Using the refined coordinates of the aligned face, the system crops a square region around it. The size of this region is determined based on facial measurements and a scaling coefficient, which is employed in the script to define the square cropping area around the detected face. This allows for the inclusion of additional space surrounding the face. The cropped image is then scaled to a standardized output size using the high-quality Lanczos interpolation method. Final images are saved in JPEG format with minimal compression settings.

The program also incorporates a critical unique face filtering stage. Following initial detection with rotation, the system filters detected faces using a two-stage approach. The first stage employs geometric filtering, where Intersection over Union (IOU) and Intersection over Area (IOA) are applied to eliminate redundant rectangles belonging to the same face. The second stage implements vector filtering, which utilizes the dlib face recognition model to compute 128-dimensional vectors (embeddings) for each face. Faces are considered unique if the distance between their vectors exceeds an established threshold (`embedding_threshold = 0.6`). This enables the system to process images containing multiple faces while preserving only one instance when faces are highly similar (e.g., from different angles).

For process monitoring and diagnostics, the system maintains detailed log files that record detection successes/failures, the detector used, and rotation angles. A separate file contains a list of images that were not processed by the Dlib detector and were passed to MTCNN.

Thus, the script implements an image processing pipeline that prioritizes the use of the Dlib detector and then, in case of its failure, switches to the backup MTCNN detector. This cascaded approach aims to optimize both detection accuracy and efficiency, since the Dlib HOG detector is

typically faster for simple cases, while MTCNN provides higher robustness in complex scenarios (e.g., poor lighting conditions, face rotations, or presence of occlusions). Through the two-stage pipeline and unique face filtering logic, the script can efficiently process images containing single or multiple faces, ensuring that each unique face is identified and processed separately.

A set of images of varying quality, orientation, and scale was used for testing. Processing was performed in Python using the `os`, `cv2`, `dlib`, `numpy`, `shutil`, `datetime`, and `MTCNN` libraries. The comparative effectiveness of the models was determined by accuracy, precision, processing time, and the success rate of face detection when changing lighting parameters, background, and their number.

As input data for testing the program's performance, two sets of photographic images were selected: the Labelled Faces in the Wild (LFW) Dataset [42] and WIDER Face Testing Images [43].

The results of processing the first dataset (13,234 photos) demonstrated the effectiveness of the tested tool: the results contained only faces, although a small portion of them were duplicated in flipped form. Several faces were partially identified on single images containing multiple faces. Among the faces that were not detected, the majority were located adjacent to other faces.

The testing results on the second dataset (16,097 photos) showed poorer performance. Most problems occurred with photographs containing groups of people positioned together but in different planes (not all faces were detected). Additionally, in photographs where perforated ribbons with text were present, as shown in Figure 5, some ribbon fragments were identified as faces.

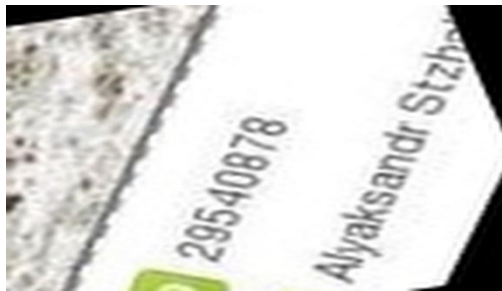


Figure 5: Sample image

Overall, it should be noted that the tool we developed has certain limitations regarding its application. For example, on images containing multiple faces, the application may not detect all of them since the system was designed for photographs containing a single central figure. Also, images containing multiple faces require substantially longer processing times, with performance scaling according to available computational resources.

The experiment confirmed that the hybrid approach of `Dlib` + `MTCNN` provides satisfactory results when working with heterogeneous images and reduces the number of missed faces.

6. Conclusions

Therefore, under conditions of full-scale and prolonged war in Ukraine, the CNN face recognition models we analyzed can be effectively utilized by law enforcement agencies to solve a wide range of tasks. These include searching for missing persons, kidnapped children, identification of deceased military personnel and civilians, detection of enemy saboteurs, collaborators and spies, war criminals and Russian military personnel, integration with video surveillance systems and drones, among others. The use of different face detector models enables deep analysis of large data volumes, particularly video materials from surveillance systems.

Within the framework of this research, recommendations have been formulated regarding the implementation of face recognition procedures and the selection of CNN models at the face detection stage to enhance the efficiency of law enforcement agencies. The developed automated system effectively solves the tasks of face detection, alignment, and cropping in images using a hybrid detector and high-quality image processing algorithms. The hybrid approach implemented

in the system allows combining the speed advantages of the Dlib detector with the enhanced robustness of MTCNN. The application of iterative rotations significantly increases the chances of face detection under non-ideal conditions. The use of Lanczos interpolation for all scaling and rotation operations minimizes image quality degradation, particularly reducing the blurriness problem. Brightness and contrast parameters have been adapted to balance between improving visibility for the detector and preserving original image details. The system's efficiency is confirmed by successful processing of a wide spectrum of images, ensuring a standardized output format.

The implemented methodologies ensure the necessary accuracy and quality of output images, making this system a valuable tool for various applied tasks, particularly in law enforcement agencies. Further research may include optimization of the scaling coefficient and integration of additional quality metrics for automatic evaluation of cropped faces. Additionally, we note that deep learning models for face recognition typically require large, suitable, labeled datasets for optimal training, which can present difficulties. Therefore, it is advisable to attempt applying the so-called ensemble method for face recognition based on deeply trained CNNs. Ensemble deep learning represents a machine learning paradigm in which several individual CNN models (learning algorithms) are combined to create a single, more effective and predictive model. Ensemble systems in face recognition for solving various tasks that law enforcement officers face in their activities today constitute the direction of our further scientific research.

Declaration on Generative AI

During the preparation of this work, the authors used OpenAI GPT-5 and Gemini in order to: Grammar and spelling check. After using these tools/services, the authors reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

- [1] A Critical Juncture amid Policy Shifts. April 2025. World Economic Outlook. <https://www.imf.org/en/Publications/WEO/Issues/2025/04/22/world-economic-outlook-april-2025>
- [2] International Monetary Fund, World Economic Outlook: A Critical Juncture amid Policy Shifts, 2025. URL: <https://www.imf.org/en/Publications/WEO/Issues/2025/04/22/world-economic-outlook-april-2025>
- [3] Office of the Prosecutor General of Ukraine, Statistical reports, 2025. URL: <https://gp.gov.ua/ua/posts/statistika>
- [4] B. Amirgaliyev, M. Mussabek, T. Rakhimzhanova, A. Zhumadillayeva, Review of ML/DL methods for person detection and face recognition, *Sensors* 25 (2025) 1410. <https://doi.org/10.3390/s25051410>
- [5] C. Deng, Review of face recognition based on deep learning, *Applied and Computational Engineering* 46 (2024) 297–303. <https://doi.org/10.54254/2755-2721/46/20241638>
- [6] X. Wang, J. Peng, S. Zhang, B. Chen, Y. Wang, Y.-H. Guo, Survey of face recognition, *arXiv* 2212.13038 (2022). <https://doi.org/10.48550/arXiv.2212.13038>
- [7] G. Guo, N. Zhang, Deep learning face recognition survey, *Computer Vision and Image Understanding* 189 (2019) 102805. <https://doi.org/10.1016/j.cviu.2019.102805>
- [8] N. El Fadel, Facial recognition algorithms: systematic review, *Journal of Imaging* 11 (2025) 58. <https://doi.org/10.3390/jimaging11020058>
- [9] K. Simonyan, A. Zisserman, Very deep convolutional networks, *arXiv* 1409.1556 (2014). URL: <https://arxiv.org/abs/1409.1556>
- [10] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, *arXiv* 1512.03385 (2015). <https://doi.org/10.48550/arXiv.1512.03385>
- [11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, et al., Going deeper with convolutions, *arXiv*

- 1409.4842 (2014). <https://doi.org/10.48550/arXiv.1409.4842>
- [12] F. Schroff, D. Kalenichenko, J. Philbin, FaceNet: unified embedding, in: CVPR, 2015. <https://doi.org/10.48550/arXiv.1503.03832>
- [13] J. Deng, J. Guo, N. Xue, S. Zafeiriou, ArcFace, in: CVPR, 2019. <https://doi.org/10.48550/arXiv.1801.07698>
- [14] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, et al., CosFace, arXiv 1801.09414 (2018). <https://doi.org/10.48550/arXiv.1801.09414>
- [15] W. Liu, Y. Wen, B. Raj, R. Singh, A. Weller, SphereFace revived, arXiv 2109.05565 (2018). <https://doi.org/10.48550/arXiv.2109.05565>
- [16] N. Zhang, J. Luo, W. Gao, MTCNN face detection, in: ICCNEA, 2020, 154–158. <https://doi.org/10.1109/ICCNEA50255.2020.00040>
- [17] J. Deng, J. Guo, E. Ververas, I. Kotsia, S. Zafeiriou, RetinaFace, in: CVPR, 2020, 5203–5212. URL: https://openaccess.thecvf.com/content_CVPR_2020/papers/Deng_RetinaFace_Single-Shot_Multi-Level_Face_Localisation_in_the_Wild_CVPR_2020_paper.pdf
- [18] N. Dakhil, A. M. Abdulazeez, Face recognition based on DL: review, Indonesian Journal of Computer Science 13 (2024). <https://doi.org/10.33022/ijcs.v13i3.4037>
- [19] X. Wang, J. Peng, S. Zhang, B. Chen, Y. Wang, Y.-H. Guo, Survey of face recognition, arXiv 2212.13038 (2022). <https://doi.org/10.48550/arXiv.2212.13038>
- [20] DCFace Authors, Balanced face generation for fair verification, arXiv 2412.03349 (2024). URL: <https://arxiv.org/abs/2412.03349>
- [21] I. D. Raji, J. Buolamwini, Actionable auditing of biased AI, in: AIES, 2019, 429–435. <https://doi.org/10.1145/3306618.3314244>
- [22] Y. Liu, J. Stehouwer, A. K. Jain, Face presentation attack detection survey, IEEE TPAMI 43 (2020) 3538–3559. <https://doi.org/10.1109/TPAMI.2020.2977021>
- [23] M. Wienroth, Socio-technical disagreements in forensic DNA, BioSocieties 15 (2020) 28–45. <https://doi.org/10.1057/s41292-018-0138-8>
- [24] M. Mordvyntsev, D. Pashniev, V. Nakonechnyi, Video analytics in criminal analysis, Law and Safety 96 (2025) 90–103. <https://doi.org/10.32631/pb.2025.1.08>
- [25] Y. P. Zaichenko, Fundamentals of Intelligent Systems Design, Slovo, Kyiv, 2004.
- [26] A. S. Dovbysh, I. V. Shelekhov, Pattern Recognition Theory, Sumy State University, 2015.
- [27] A. S. Dovbysh, Intelligent Systems Design, Sumy State University, 2009.
- [28] OpenCV, Cascade Classifier, 2025. URL: https://docs.opencv.org/4.x/db/d28/tutorial_cascade_classifier.html
- [29] C. Antipona, R. Magsino, Haar cascade enhancement for face recognition, 2024. <https://doi.org/10.13140/RG.2.2.34675.75045>
- [30] H. G. Shah, V. B. Suthar, S. P. Thakkar, V. M. Thumar, Face detection on Raspberry Pi, IRJAEH 2 (2024) 2440–2445. <https://doi.org/10.47392/IRJAEH.2024.0334>
- [31] C.-L. Lin, Y.-H. Huang, Adaptive facial attendance systems, Electronics 11 (2022). <https://doi.org/10.3390/electronics11142278>
- [32] M. Zamir, N. Ali, A. Naseem, A. Frasteen, B. Zafar, M. O. Assam, Face recognition on Raspberry Pi, Computation 10 (2022) 148. <https://doi.org/10.3390/computation10090148>
- [33] J. Deng et al., RetinaFace: dense face localisation, arXiv 1905.00641 (2019). <https://doi.org/10.48550/arXiv.1905.00641>
- [34] D. Qi, W. Tan, Q. Yao, J. Liu, YOLO5Face, arXiv 2105.12931 (2021). <https://doi.org/10.48550/arXiv.2105.12931>
- [35] D. Brown, Mobile attendance using OpenVINO, in: ICAIS, 2021, 1152–1157. <https://doi.org/10.1109/ICAIS50930.2021.9395836>
- [36] S. Demirkol (serengil), DeepFace, GitHub repository, 2025. URL: <https://github.com/serengil/deepface>
- [37] N. Dalal, B. Triggs, HOG for human detection, in: CVPR, 2005, 886–893. <https://doi.org/10.1109/CVPR.2005.177>
- [38] K. Zhang, Z. Zhang, Z. Li, Y. Qiao, MTCNN, IEEE SPL 23 (2016) 1499–1503.

<https://doi.org/10.1109/LSP.2016.2603342>

- [39] K. J. Zuiderveld, CLAHE, in: Graphics Gems IV, 1994, 474–485. <https://doi.org/10.1016/B978-0-12-336156-1.50061-6>
- [40] V. Kazemi, J. Sullivan, One millisecond face alignment, in: CVPR, 2014, 1867–1874. <https://doi.org/10.1109/CVPR.2014.241>
- [41] R. C. Gonzalez, R. E. Woods, Digital Image Processing, 4th ed., Pearson, 2018.
- [42] LFW Dataset, Kaggle, 2025. URL: <https://www.kaggle.com/datasets/jessicali9530/lfw-dataset>
- [43] WIDER Face Dataset, 2025. URL: <http://shuoyang1213.me/WIDERFACE/>