Face Detection for Video Surveillance-based Security System

Olena Yakovleva[1,3], Andrii Kovtunenko[2,3], Valentyn Liubchenko[2,3], Vadym Honcharenko[3] and Oleg Kobylin[2]

[1] *Bratislava University of Economics and Management, Furdekova 16, Bratislava, 85104, the Slovak republic*
[2] *Kharkiv National University of Radio Electronics, Nauky Ave. 14, Kharkiv, 61166, Ukraine*
[3] *SYTOSS s.r.o., Bratislava, Vajnorská 10645/100, Bratislava, 831 04, the Slovak republic*

Abstract

This paper is devoted to the study of face detection methods MTCNN, FaceBoxes, DSFD, RetinaFace, CenterFace, and SCRFD and aims to select the most suitable one for an enterprise security system based on employee face recognition from video surveillance cameras. The time costs of these methods and their robustness to geometric scale distortions and rotations in different planes have been analyzed. For the experiments, custom datasets have been created. Particular attention has been paid to the compromise between the speed and accuracy of the considered methods for their use as the first stage of a security system based on face recognition in a video stream. The conducted research has shown that RetinaFace-MobileNet0.25, FaceBoxes, SCRFD-500MF, CenterFace were the fastest; RetinaFaceResNet125, DSFD, RetinaFaceMobileNet0.25 were the most robust to face rotation; almost all models were robust to changes in the face size. Also, when choosing the most suitable face detection method to apply for a security system, the presence of landmarks was taken into account, as well as the fact that the recognition methods used at the next stage have their limitations in terms of robustness to changes in scale and rotation. Considering the above requirements, it was decided to use the RetinaFaceMobileNet0.25 method in the pipeline of the security system as the first stage for face detection.

Keywords

face detection, embeddings, landmarks, MTCNN, FaceBoxes, DSFD, RetinaFace, CenterFace, SCRFD, security system, video surveillance

1. Introduction

The sphere of computer vision is advancing fast, just like technical and scientific progress in general. New challenges appear and stack with existing tasks, which, in turn, require new and effective solutions. These new challenges are dedicated to a specific industry and problem, for example, deep learning replaced classical methods of computer vision [1–4] because there was a need to improve the quality of the classification task, and at that time, classical methods did not solve it with sufficient quality. The same can be said about the identification task, namely, face detection and recognition. The problem of protecting property, an object or other resource is not new and is still relevant. Various sets of measures, tools and security systems are used to solve it. The development of the last listed security systems was facilitated by the development of computer vision because it is much more convenient to install video cameras and automatically analyze what is happening. For example, a similar scenario is implemented at airports to provide security. Airport security systems analyze people's faces and look for them in search databases, analyzing emotions, human conditions, and many other characteristics.

COLINS-2023: 7th International Conference on Computational Linguistics and Intelligent Systems, April 20-21, 2023, Kharkiv, Ukraine
EMAIL: olena.yakovleva@sytoss.com (O. Yakovleva); andrii.kovtunenko@nure.ua (A. Kovtunenko); valentyn.liubchenko@nure.ua (V. Liubchenko); vadim.honcharenko@sytoss.com (V. Honcharenko); oleg.kobylin@nure.ua (O. Kobylin)
ORCID: 0000-0002-6129-6146 (O. Yakovleva); 0009-0004-9072-7779 (A. Kovtunenko); 0000-0002-9966-0249 (V. Liubchenko); 0009-0002-0370-3361 (V. Honcharenko); 0000-0003-0834-0475 (O. Kobylin)

Face detection is the first and critical step in face analysis and recognition, and the quality and speed of this first step determine all subsequent steps taken to solve various security tasks. Many face detection methods have already been created, and even more will be created in the future because, as mentioned earlier, applications, tasks and conditions change. One method is applied to one environment and needs, and another technique suits another field. Therefore, for a specific face analysis algorithm, it is necessary to have arguments regarding the choice of the best-fitting detection method to be used at the first step of a complex pipeline.

The purpose of the study was to investigate methods for face detecting and finding the best in terms of accuracy and time costs, as well as the best meeting the requirements of a video surveillance-based security system.

Thus, the object of this study paper is the issue of face recognition in a video stream from surveillance cameras. The subject of the study is the properties of face detection methods that serve as the first stage of recognition, namely the possibility of their use in real time, their robustness to rotation and face scaling.

2. Current State of Face Detection

The first face detection methods were based on classical approaches in which features specified by researchers were extracted from an image or its part: the presence of eyes, nose, mouth, their relative position, and pattern search. Later, there appeared methods that extracted structural features from images and transferred those features to the classifier in the next step: the Viola-Jones method [5], which analyzes images with a set of Haar primitives in a sliding window, and the analysis of the histogram of oriented gradients (HOG) [6]. Such methods significantly improved detection at that time and are relevant today. Still, their accuracy can no longer compete with newer approaches, significantly when the detection conditions change, and faces are partially overlapped.

The rapid growth of detection quality coincided with the emergence of deep neural network architectures. The advantage is that a network trained for multi-class classification can detect the necessary features that characterize a face independently and find it. The first such set of methods used a cascade approach, where several networks scanned the image at once, and the next set of networks further scanned the selected regions. Such an approach improved the detection quality in advance, but the detection speed depended on the number of faces in the image [7].

The approaches that used the ideas of R-CNN [8] and Faster-RCNN [9] methods became the next stage of evolution because, at that time, they were leaders in solving the object detection problem. In these methods, a separate network module Region Proposal Network searches for candidates and simultaneously selects parts of the image. These parts are then transferred to the classifier for the final conclusion about the presence of a face. Starting with Faster-RCNN, training was performed for the entire network instead of separately for each module, making those methods advantageous. At the same time, the main disadvantage of this approach was speed. Therefore, techniques appeared that performed detection in one stage. They were complemented by methods based on feature pyramids [10]. This breakthrough made it possible to combine semantically weak features with semantically strong ones. In turn, these achievements improved the accuracy of detecting faces of small size, with differences in brightness or partially overlapped with other objects.

Another group of methods is those based on heat maps produced by a convolutional network. That means the method does not return the exact coordinates of the face location but instead gives the pixel-by-pixel probability of a pixel belonging to a face class. The evolution of detection methods is not complete at this point, and studies are still underway to achieve better results in the face detection task.

The process of object detection is of primary importance in solving many computer vision problems. Although face detection is already considered to be a sufficiently solved problem, it still needs further research to adapt to the specific conditions of a particular task.

In this paper, the detection issue is analyzed to solve the problem task of face recognition for a medium-sized enterprise security system based on processing a stream from video surveillance cameras.

In this paper, the detection problem is analyzed to solve the task of face recognition for a medium-sized enterprise security system based on processing a stream from video surveillance cameras. In this paper, a medium-sized enterprise is defined as an enterprise with up to 400 employees.

There are two main aspects of a security system: speed and accuracy. Meanwhile, accuracy should be considered simultaneously for the entire system and each of its separate modules. A security system must be able to quickly detect people in the field of view under various conditions, record these events into a log, and recognize people in parallel. Recognition should occur in real time. A person should not have to perform additional actions to be recognized. All the operations should take place in the background. If the system recognizes a person, the front door should open automatically, otherwise a security guard will be called. The system should be easily configurable and capable of processing many cameras simultaneously.

Thus, the following requirements have been set for the system:

- process one frame in no more than 100 ms;
- be insensitive to changes in the position of cameras and people, and changes in lighting;
- detect faces no smaller than 100×100 px;
- notify about a decrease in the quality of recognition;
- the quality of the system must be at least 98% according to the AUC ROC value.

3. Face Detection Methods and Their Analysis

The study of face detection methods aims to choose the most suitable one for an enterprise security system based on recognizing employees from video surveillance cameras. According to the security system requirements, special attention was paid to the compromise between the speed and robustness of the analyzed methods for their use as the first stage of the security system. After researching the sources, the following trained neural networks models were selected for analysis (Table 1):

- Multi-Task Cascaded Convolutional Networks (MTCNN) [11];
- FaceBoxes [12];
- Dual Shot Face Detector (DSFD) [13];
- RetinaFace [14];
- CenterFace [15];
- Single-stage Cascade Residual Face Detector (SCRFD) [16].

Table 1 provides a short description of the studied models. RetinaFace was considered in two variants, using Resnet50 and Mobilenet0.25 as backbone networks. For the SCRFD network, we chose the SCRFD-500MF model, which costs 500 Mega FLOPs for input images in VGA (640x480) resolution.

Table 1

Brief information about the detection models

Method name	Year of appearance	Availability of landmarks	AP on a validation set for WIDER FACE (Hard)
MTCNN	2016	+	0,598
FaceBoxes	2018	-	0,395
DSFD	2018	-	0,904
RetinaFaceResNet125	2019	+	0,918
RetinaFaceMobileNet0.25	2019	+	0,78
CenterFace	2020	+	0,875
SCRFD-500MF	2021	-	0,685

Each of the neural networks has its characteristics, which are given in their primary papers [11–16]. MTCNN uses a cascade of three convolutional neural networks to detect faces in images. It is especially well suited for detecting faces of different sizes and orientations [11]. The main feature of the FaceBoxes model is its ability to detect faces at high speed. This ability makes it ideal for applications requiring high processing speed, such as security systems [12]. DSFD is a model that uses two stages for face detection. It has two branches of convolutional neural networks, each identifying areas with different scales and detecting faces in corresponding areas. This approach allows this model to detect

faces of different sizes and angles in images [13]. The headline characteristic of the RetinaFaceResNet125 model is that it uses two convolutional neural networks: RetinaFace for face detection and ResNet125 for feature extraction. This way, it detects faces with high accuracy and determines more detailed information about faces in images. RetinaFaceMobileNet0.25 is a model similar to RetinaFaceResNet125, but instead of ResNet125, it uses the lighter MobileNet0.25 network, making it more applicable for use on mobile devices [14]. CenterFace is a model that uses the center point of a face to detect its position in an image. It is capable of detecting small and rotated faces in images [15]. SCRFD is a model that uses multiple convolutional layers to detect faces in images. It features a high processing speed and can detect faces of different sizes in images [16].

Average precision (AP) is often used to compare the accuracy of detection methods. Average precision (AP) is the area under the Precision-Recall curve, which can be calculated as an approximation of the Precision-Recall curve with rectangles as follows [17]:

$$AP = \sum_{k=0}^{n-1} \big(r(k) - r(k-1)\big) * \max\,(p(k), p(k-1)), \quad (1)$$

where n is the number of threshold values compared with which the confident values that the method returns as a detection result, $k = 0, \ldots, n-1$ (these thresholds are used to build the Precision-Recall curve); $r(k)$ – the Recall value for the threshold k; $p(k)$ – the Precision value for the threshold k.

In this study, AP was not calculated but was taken from the following primary publications and repositories where models are in the public domain: [15] for MTCNN, FaceBoxes, DSFD, CenterFace; [14, 15] for RetinaFace; [16, 18] for SCRFD-500MF. For all models, the AP was calculated on the WIDER FACE (Hard) validation dataset [19].

All these networks can work with faces of different sizes and changing face angles. However, the authors of this paper have yet to find any studies that describe the range of scale factor values and the angles of face rotation within which the methods have the declared accuracy.

Also, the speed of the models presented in the existing sources was measured on different devices and for different image sizes, which makes it impossible to compare the time costs of these models. For example, in [11], the speed of the MTCNN method was 16 FPS (2.60GHz CPU) and 99 FPS (Nvidia Titan Black GPU); the speed of FaceBoxes was 20 FPS (Intel Xeon E5-2660v3@2.60GHz) [12]; the RetinaFaceResNet125 model was 13 FPS for VGA images (640×480 px), the RetinaFaceMobileNet0.25 – 40 FPS (4K images (4096×2160 px) at NVIDIA Tesla P40 GPU), 20 FPS (HD images (1920×1080 px) at Intel i7- 6700K CPU), 16 FPS (VGA images at ARM-RK3399) [14]. In the work [15], there is a comparison of the operating speed of DSFD and CenterFace models on NVIDIA GTX2080TI, where the following results are obtained: 78.08 ms and 5.51 ms for VGA-resolution images, respectively. It is also shown that the CenterFace model demonstrates a speed of 30 FPS on the CPU I7-6700@2.6 with state-of-the-art accuracy. Paper [16] determine that SCRFD-0.5GF outperforms RetinaFaceMobileNet0.25 by 45.57% in time. The DSFD method has a very low speed that cannot be used without GPU.

The information above allows us to conclude that the SCRFD-0.5GF model can be claimed as the fastest of the considered models, and the DSFD is most likely the slowest. However, the matter of speed requires further investigation. It is necessary to measure the speed for MTCNN, FaceBoxes, DSFD, RetinaFaceResNet125, RetinaFaceMobileNet0.25, CenterFace, and SCRFD-500MF models on the same device for images of the same size.

3.1. Purpose and Content of Experiments

Since the purpose of this study is to select a detection method to be used at the first stage of a face recognition system, i.e., its result will significantly affect the final recognition result, it is necessary to investigate the following properties of detection methods:

- robustness to rotations in different planes (angle changes around the vertical and horizontal axes from -90 to 90 degrees);
- robustness to geometric scale distortions (changing the size of face images from 20×20 to 310×310);
- time costs (for face detection in two sizes of 640×480 (VGA) pixels and 1280×720 (HD) pixels).

In the experiments, we used neural models from publicly available repositories: MTCNN [20], FaceBoxes [21], DSFD [22], RetinaFace [22], CenterFace [23], SCRFD-500MF [24]. The confidence values returned by these models were used to evaluate the detection quality.

The experiments were conducted on the custom datasets created using the Generated [25] and Blender editors [26].

The time consumption was measured on an NVIDIA GeForce 940MX mobile graphics card.

3.2. Dependence of Detection Quality on Face Rotation

To compare the methods for the detection quality at different angles of human head rotation, we created our custom dataset, in which human faces were taken using Generated Photos [25], and 3D models were created based on these images. Examples of images from the generated dataset are shown in Fig. 1.

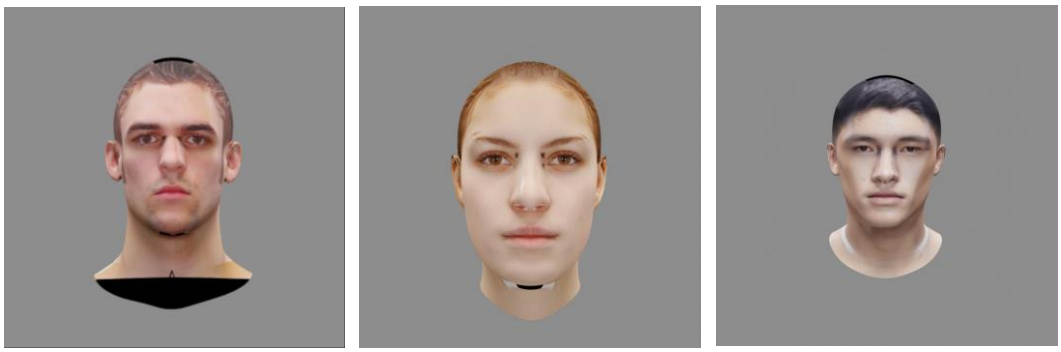

Figure 1: Examples of images created with the Generated Photos editor

This 3D model was rendered in the graphic editor Blender [26] with a rotation step of one degree in two axes separately from left to right from -90 degrees to +90 degrees (rotation around the X axis) and an elevation head angle from bottom to top from -90 degrees to +90 degrees (rotation around the Y axis). That means that 181 images were generated for each 3D model. Fig. 2 shows examples of images from the synthesized dataset of the got 3D models.

Figure 2: Examples of images from an artificial dataset created using the Blender editor: on the left – rotation around the Y-axis by -50 degrees (up-down motion); on the right – rotation around the X-axis by -20 degrees (left-right motion)

Fig. 3 and 4 show the values of average confidence in the case of rotation around the Y-axis and in the case of the X-axis, respectively.

The final Table 2 contains the results of the experiments and lists the ranges of changes in the angle values within which the confidence was greater than 0.9.

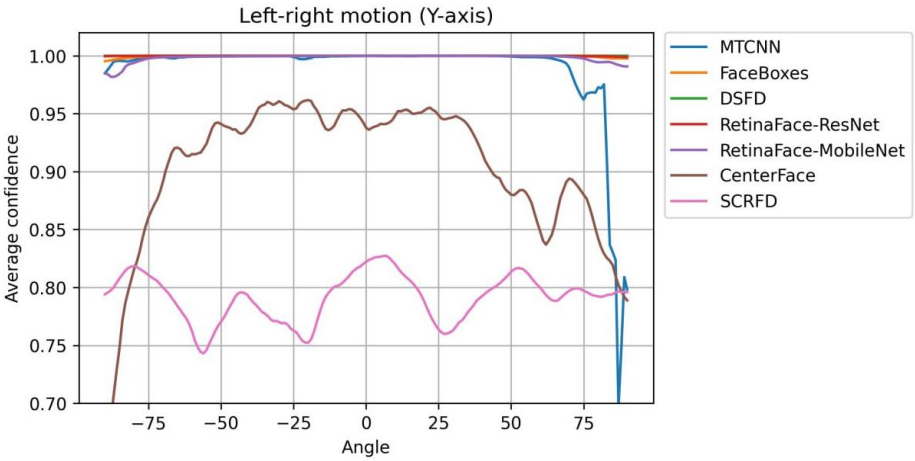

Figure 3: Dependence of detection quality on the angle of rotation around the Y-axis (left-right motion)

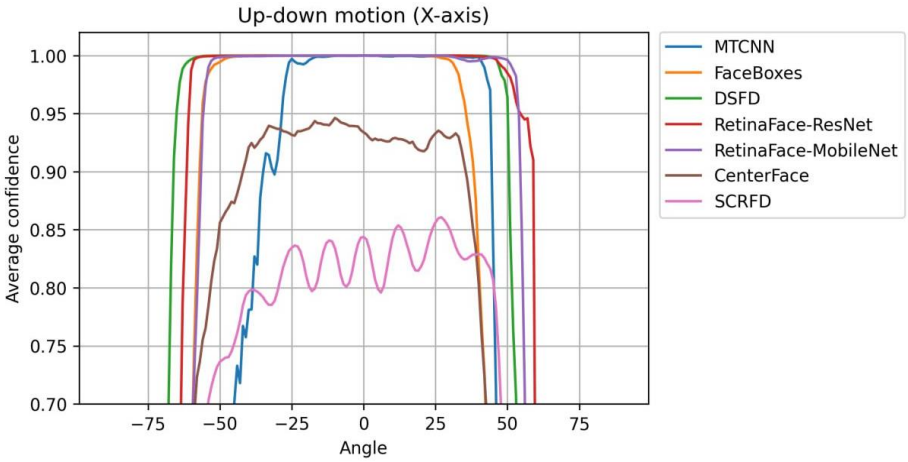

Figure 4: Dependence of detection quality on the angle of rotation around the X-axis (up-down motion)

Table 2

Experimental results on detection accuracy under face rotation conditions

Method	Range of rotation angles around the Y-axis (left-right motion), average confidence ≥ 0.9	Range of rotation angles around the X-axis (up-down motion), average confidence ≥ 0.9
MTCNN	[-88;82] (range=171)	[-30;46] (range=77)
FaceBoxes	[-90;90] (range=181)	[-60;38] (range=99)
DSFD	[-90;90] (range=181)	[-69;50] (range=120)
RetinaFaceResNet125	[-90;90] (range=181)	[-63;60] (range=124)
RetinaFaceMobileNet0.25	[-90;90] (range=181)	[-60;55] (range=116)
CenterFace	[-70;45] (range=166)	[-40;35] (range=76)
SCRFD0.5GF	∅	∅

In the case of rotations around the Y-axis (left-right motion), the average confidence value above 0.9 and the maximum possible range of angle changes [-90;90], or range=181, are consistently shown by FaceBoxes, DSFD, RetinaFaceResNet125, RetinaFaceMobileNet0.25 models. MTCNN detects with an average confidence value higher than 0.9 in the range [-88;82]. Next comes the CenterFace model with a range significantly less than [-70;45] and average confidence not higher than 0.97. The most unstable model was SCRFD-500MF, whose average confidence varied in a sinusoidal manner by 0.7 values and did not exceed values above 0.83. In addition, FaceBoxes, DSFD, and

RetinaFaceResNet125 have an average confidence value close to 0.99 over the entire range. The RetinaFaceMobileNet0.25 and MTCNN models have average confidence close to 0.99 in the range [-75;75] and [-77;70], respectively.

Regarding rotation around the X-axis (up-down motion), the FaceBoxes, DSFD, RetinaFaceResNet125, and RetinaFaceMobileNet0.25 models also showed better results. However, around the X-axis, the range of rotation angle values was much smaller compared to the Y-axis. In the experiments with average confidence above 0.9, the best models were RetinaFaceResNet125, DSFD, RetinaFaceMobileNet0.25 with ranges of 124, 120, 116, respectively, followed by FaceBoxes with a range of 99. The MTCNN and CenterFace models have a similar range of 77 and 76, respectively, but average confidence for CenterFace does not exceed 0.95. The average confidence value of SCRFD-500MF changes unstably, the graph is sinusoidal and does not exceed 0.87. It can also be noted that all the considered models, except CenterFace, SCRFD-500MF, achieve average confidence values close to 0.99 (DSFD, RetinaFaceResNet125, RetinaFaceMobileNet0.25 in the range [-50;47], FaceBoxes – [-48;25], MTCNN – [-20;40].

In general, in terms of detection accuracy under the presence of face rotation, the considered methods can be ranked as follows: RetinaFaceResNet125, DSFD > RetinaFaceMobileNet0.25 > FaceBoxes > MTCNN >> CenterFace >> SCRFD-500MF, where ">" indicates that the method on the left is more robust than the one on the right, ">>" means that there is a significant difference between the robustness of the methods.

3.3. Dependence of Detection Quality on Face Size

An additional artificial dataset was created to study the accuracy of face detection methods depending on face size. First, 10 faces were generated using the Generated Photos editor [https://generated.photos/]. In the next step, test images were created for each face to present them in different sizes. The face size was changed in 10-pixel increments. The smallest face size is 20×20 pixels, the largest is 310×310 pixels. For detection models that use non-maximum suppression (NMS) to combine results, a threshold of 0.3 was set for NMS. The images were transferred to the methods without additional normalization or resizing.

Fig. 5 and 6 show some examples of image detection results from an artificial dataset. The detected faces are surrounded by a box, the color of which depends on the confidence value returned by the detection model, and the confidence value is also displayed next to it in figures.

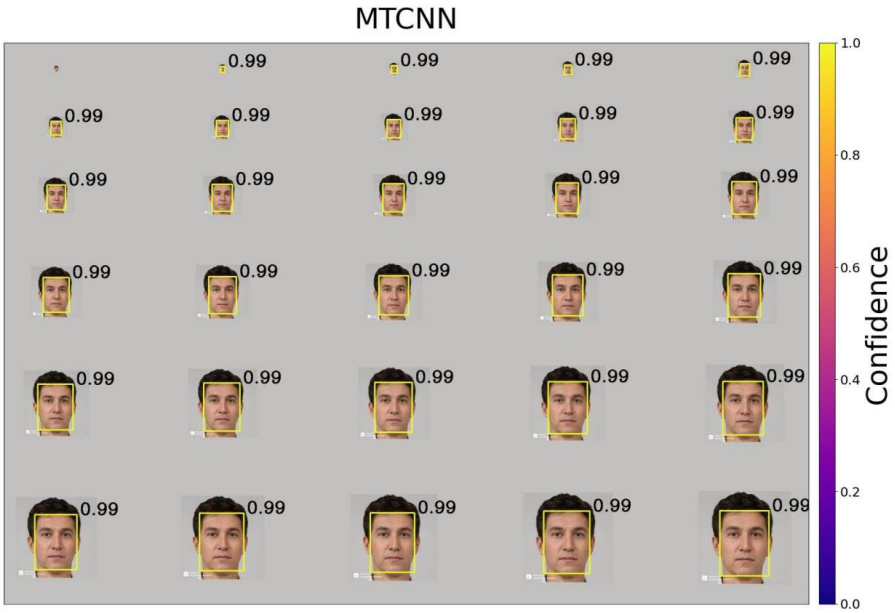

Figure 5: Dependence of confidence values for MTCNN on an image size that varies from varies from 20×20 to 310×310 (px)

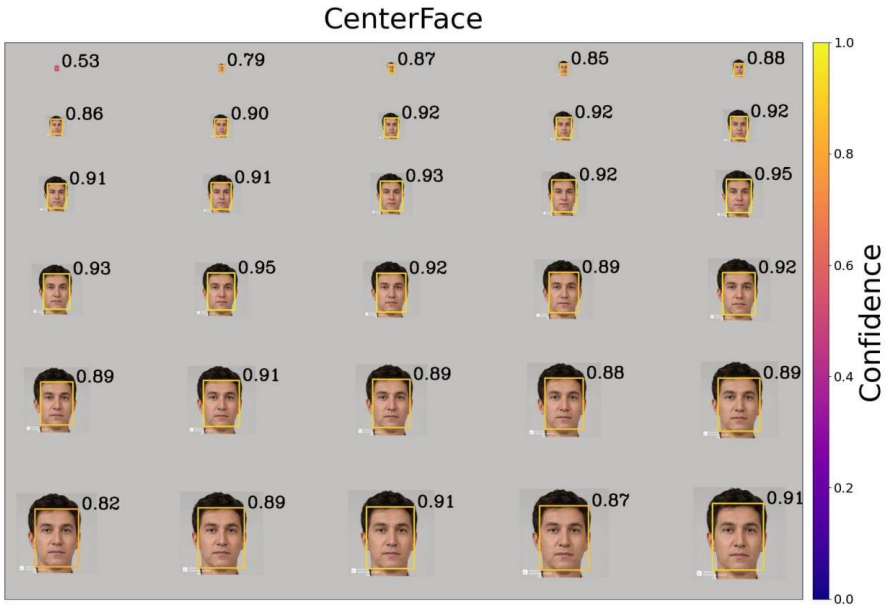

Figure 6: Dependence of confidence values for CenterFace on an image size that varies from 20×20 to 310×310 (px)

The average confidence values were calculated for each face size based on the confidence values for all experiments. Fig. 7 shows the dependence of average confidence on the change in face size.

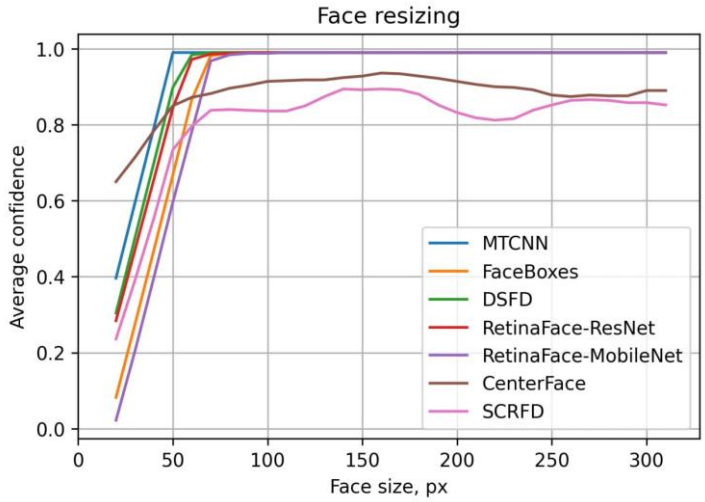

Figure 7: Dependence of the average confidence values on the change in image size from 20×20 to 310×310 (px)

It can be concluded that the MTCNN, FaceBoxes, DSFD, RetinaFaceRes-Net125, RetinaFace-MobileNet0.25 models detect faces with sizes from 75×75 to 310×310 very stably (average confidence ≥ 0.99). In addition, the MTCNN model produces average confidence ≥ 0.99 starting from a face size of 50×50, and in some cases, it detected faces of 30×30 with confidence 0.99. That means that MTCNN detects the smallest face size with high confidence, often even 30×30 faces, as shown in Fig. 6, have average confidence ≥ 0.99. For DSFD, RetinaFace-ResNet125, the size of the first face should be at least 60 to get average confidence ≥ 0.99. The FaceBoxes, RetinaFace-MobileNet0.25 models have average confidence ≥ 0.99 for image sizes starting from 70×70. The CenterFace model does not have average confidence higher than 0.95 over the entire range but shows the highest average confidence of 0.62 for very small images of 20×20. To compare, for images of this size, the MTCNN, RetinaFace-ResNet125, FaceBoxes, DSFD models have much lower average confidence. CenterFace has average confidence higher than 0.90, only starting from 80×80 pixels. Still, this method was able to find the

smallest face of 20×20 and has a smaller scatter of probabilities with rescaling. The SCRFD0.5GF model detects images starting at 30×30, but the average confidence does not reach above 0.9, and it has 0.9 only in a small range of sizes from 140×140 to 170×170. Moreover, the model showed unstable operation during experiments.

For visual clarity, Table 3 lists the dimensions of the minimum face size with average confidence greater than 0.9, and 0.99.

Table 3
Results of research on detection methods under the conditions of face size changes

Method	The minimum face size (px), average confidence ≥ 0.9	The minimum face size (px), average confidence ≥ 0.99
MTCNN	45×45	50×50
FaceBoxes	60×60	70×70
DSFD	50×50	60×60
RetinaFace-ResNet125	53×53	60×60
RetinaFace-MobileNet0.25	63×63	70×70
CenterFace	80×80	∅
SCRFD0.5GF	140×140	∅

To conclude, to rank the methods in terms of the largest range of face size with an average confidence of 0.99 or higher, we'll get the following sequence: MTCNN>DSFD, RetinaFace-ResNet> FaceBoxes, RetinaFace-MobileNet0.25>> CenterFace> SCRFD-500MF.

3.4. Time Costs Comparison

The methods were compared in terms of time costs on two image sizes: VGA (640×480 pixels) and HD (1280×720 pixels). The reason is that these sizes are common for cameras, and as demonstrated in previous tests, detection can occur on faces as small as 20 pixels. So, it is necessary to study the performance of networks under different conditions since it is possible to reduce or increase the frame resolution and, this way, to change the time costs in one direction or another. All measurements were made with a preliminary "warming-up" of the network, skipping the first detection, on a mobile video card NVIDIA GeForce 940MX.

The results of the speed measurements of face detection for 100 images are shown in the diagrams, where the X-axis is the time value (ms, on a logarithmic scale), and the Y-axis is the normalized number of frames with the certain frame processing time (Fig. 8 and 9).

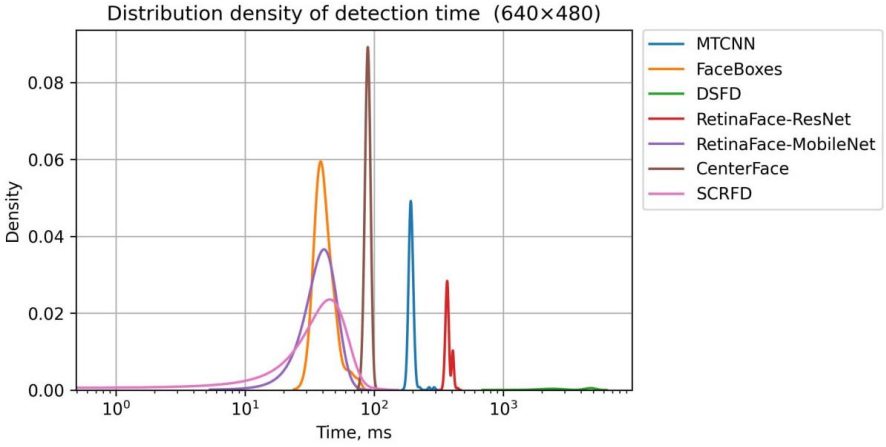

Figure 8: Distribution density of frame processing time for VGA images (X-axis is logarithmic)

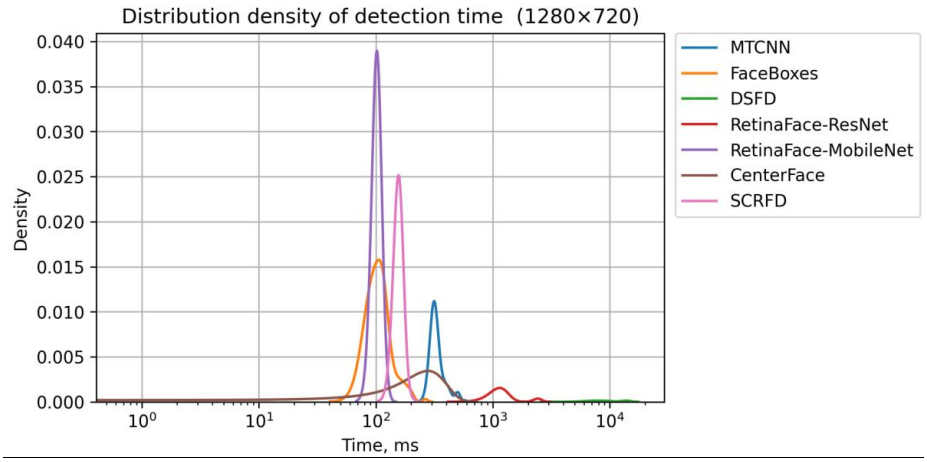

Figure 9: Distribution density of frame processing time for HD images (X-axis is logarithmic)

Fig. 10 is a graph showing the basic time measurement statistics for 640×480 images for all methods, such as medians, first 0.25% and third 0.75% quartiles, and outliers. The statistical patterns for 1280×720 images are similar. However, the DSFD method has an even greater scatter, and the RetinaFace-ResNet125 method has a 26-fold increase in scatter (Fig. 11).

Table 4 collects the numerical values of such statistical values as average time, median, and standard deviation of face detection time for images of 640×480 and 1280×720.

Table 4
The results of detection time measurements

Method	Average frame processing time, ms 640×480	Median of frame processing time, ms 640×480	Standard deviation of frame processing time, ms 640×480	Average frame processing time, ms 1280×720	Median of frame processing time, ms 1280×720	Standard deviation of frame processing time, ms 1280×720
MTCNN	195	193	14	345	317	59
FaceBoxes	44	41	9	115	109	32
DSFD	2866	2300	1007	8758	7092	2804
RetinaFace-ResNet	377	371	19	1375	1178	496
RetinaFace-MobileNet	42	40	5	103	102	5
CenterFace	89	89	1	279	274	16
SCRFD-500MF	46	44	8	157	155	8

Coefficients reflecting the increase in detection time when upscaling from VGA (640×480 pixels) to HD (1280×720 pixels) were also calculated:

$$Avg\_time\_increase = avg\_time\_HD/avg\_time\_VGA, \quad (2)$$

$$Std\_time\_increase = std\_time\_HD/std\_time\_VGA, \quad (3)$$

where avg_time_HD – is the average frame processing time for HD images; avg_time_VGA – average frame processing time for VGA images; std_time_HD – standard deviation of frame processing time for HD images; std_time_VGA – standard deviation of frame processing time for VGA images.

Thus, the coefficients $Avg\_time\_increase$ and $Std\_time\_increase$ illustrate the rate of growth of average frame processing time and standard deviation of frame processing time when increasing the image size from VGA to HD.

The calculated coefficients (2) and (3) are shown in Fig. 12.

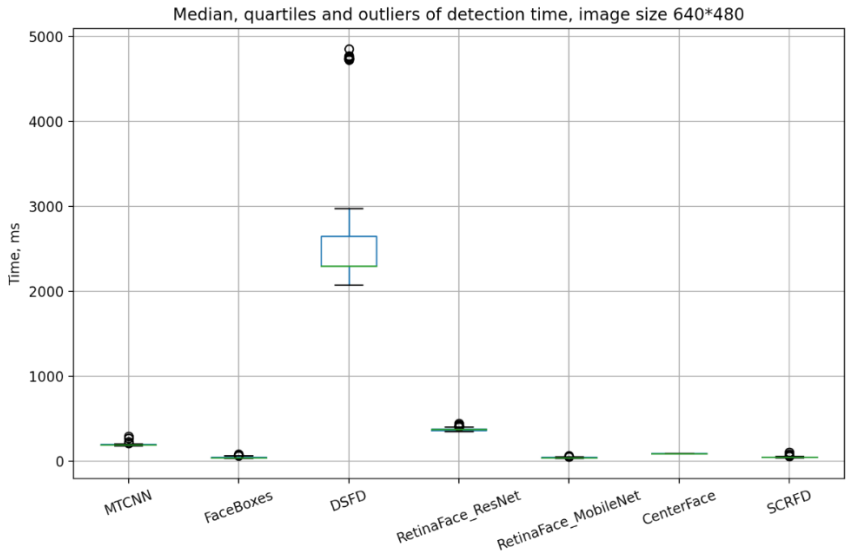

Figure 10: Frame processing time statistics for VGA images (X-axis is logarithmic)

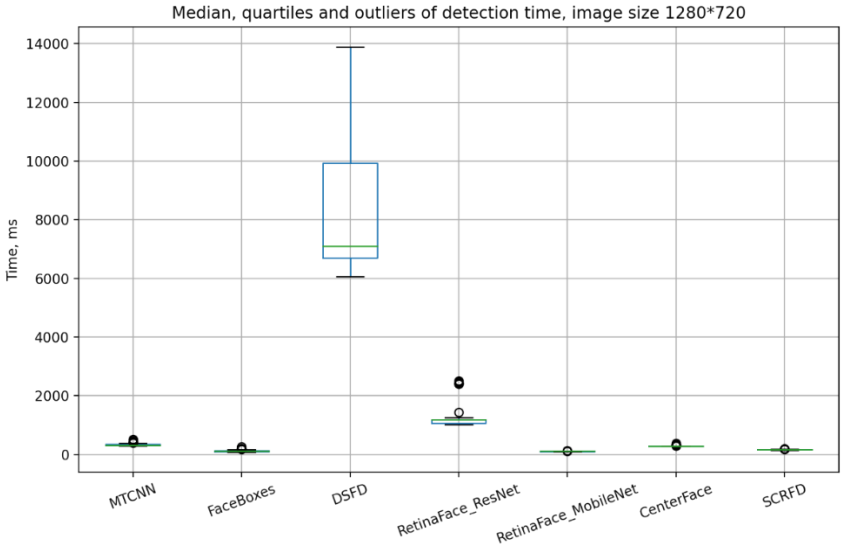

Figure 11: Frame processing time statistics for HD images (X-axis is logarithmic)

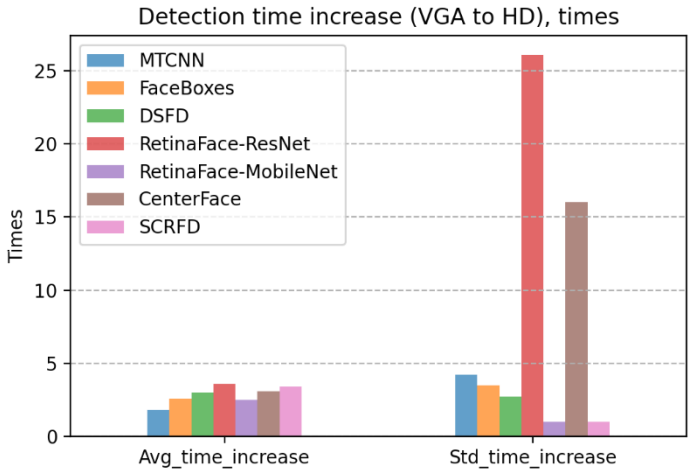

Figure 12: Increase in average frame processing time, standard deviation of frame processing time (from VGA to HD)

Based on Fig. 8–12 and the data in Table 4, it can be noted that the fastest models are RetinaFace-MobileNet0.25, FaceBoxes, and SCRFD-500MF, as they have similar average frame processing times, which are 42 ms, 44 ms, and 46 ms for 640×480 px images, and 103 ms, 115 ms, and 157 ms for 1280×720 px images, respectively. Moreover, the variation of the average frame processing time for these models is very small from experiment to experiment. Next is the CenterFace model, which has detection times of 89 ms and 274 ms for 640×480 px and 1280×720 px images, respectively. This methid is 2.1 times faster than RetinaFace-MobileNet0.25 for 640×480 and 2.7 times faster for 1280×720. The detection time of the MTCNN model is 2.2 times faster than CenterFace for 640×480 images but only 1.2 times faster for 1280×720 images. With RetinaFace-ResNet125 face detection takes even more time. This network already works 9 times longer than RetinaFace-MobileNet0.25 for 640×480 and 11.5 times longer for 1280×720. The DSFD method is significantly different from the others, it detects 68 times slower than RetinaFace-MobileNet0.25 for 640×480 px and 85 times slower for 1280×720 px. In addition, DSFD has a large scatter compared to other methods, which is 1007 ms (640×480). For example, the scatter for the CenterFace method is 1 ms, RetinaFace-ResNet125 – 19 ms.

When upscaling images from VGA (640×480=307200 pixels) to HD (1280×720=921600 pixels), the number of pixels in the image increases by 3 times. The processing time increases similarly, $Avg\_time\_increase$ from 1.8 for MTCNN to 3.6 for RetinaFace-ResNet125. However, regarding the amount of scatter, RetinaFace-MobileNet0.25 has a much larger scatter for HD images, Std_time_increase to 26 times. This means that for RetinaFace-MobileNet0.25, the image size significantly impacts on the stability of the processing time.

Based on the experiments, the considered methods can be ranked by time costs as follows: RetinaFace-MobileNet0.25, FaceBoxes, SCRFD-500MF > CenterFace > MTCNN > RetinaFace-ResNet125 >>> DSFD.

3.5. Experiments Conclusions and Detection Method Choice for Security System

The conducted experiments on face detection with the presence of face rotations, changes in their size, and speed measurement allow us to draw the following conclusions in a comparative aspect regarding the methods MTCNN, FaceBoxes, DSFD, RetinaFaceResNet125, RetinaFaceMobileNet0.25, CenterFace, and SCRFD-500MF.

All models showed a higher robustness to face rotation around the Y-axis (left-right motion) than around the X-axis (up-down motion); so around the Y-axis, detection occurred for a wider range of angles. However, when rotating around both axes, the RetinaFaceResNet125, DSFD, RetinaFaceMobileNet0.25 models work stably with an average confidence of about 0.99 under rotation conditions [-45;45]. The FaceBoxes model also shows average confidence close to 0.99 when rotating around both axes but has a smaller range. It should be noted that all models except SCRFD-500MF show average confidence of more than 0.9 in the range [-35;35]. The average confidence of SCRFD-500MF does not exceed 0.87 (see more details in 3.2).

In the experiments with changes in faces size, the MTCNN, DSFD, RetinaFace-ResNet, FaceBoxes, and RetinaFace-MobileNet0.25 models performed better than the CenterFace and SCRFD-500MF models. MTCNN detected the smallest face size of 50×50, with average confidence close to 0.99. Face size from 70×70 is consistently processed by MTCNN, DSFD, RetinaFace-ResNet, FaceBoxes, RetinaFace-MobileNet0.25 models, with average confidence ≥ 0.99, and from 63×63 — with average confidence ≥ 0.9. CenterFace, SCRFD-500MF models detect with average confidence ≥ 0.9 starting from 80×80 and 140×140, respectively. The SCRFD-500MF model does not achieve average confidence higher than 0.9. In some cases, the CenterFace and SCRFD0.5GF models detect very small faces of 20×20 px and 30×30 px, respectively (see more details in 3.3).

The fastest methods were RetinaFace-MobileNet0.25, FaceBoxes, and SCRFD-500MF, with an average processing time of less than 47 ms for VGA images. For HD images, the RetinaFace-MobileNet0.25 and FaceBoxes models have about 100 ms, meanwhile, the SCRFD-500MF model shows 155 ms. The CenterFace model shows a time of less than 100 ms for VGA images, but for HD images, it shows a much longer time. The MTCNN, RetinaFace-ResNet125 models exceed 100 ms

even for VGA images. The DSFD model cannot be used in real-time because it has 1007 ms for VGA images. In addition, the DSFD and RetinaFace-ResNet125 models have significant scatters in processing times for one size images (see 3.3 for more details).

To facilitate and make it easier to choose the detection method that best meets the security requirements, it was decided to present information about the specific properties of the methods in a convenient summary form. The following properties were selected as the most important:

- AP – accuracy of the method (Table 1);
- Landmarks – the presence of landmarks (landmarks are used to normalize faces before they are recognized, i.e., the ability of the model to provide landmarks is beneficial and allows to avoid spending additional time calculating landmarks after detection) (Table 1);
- MaxFaceAngle – the maximum range of rotation angle for which the method detects faces with a confidence of more than 0.9 (must correspond to both axes) (Table 2);
- MinFaceSize – the minimum face size that can be detected with a confidence of more than 0.9 (Table 3);
- DetectionTime – the average frame processing time of one frame for VGA images (Table 4).

All numerical values of the properties listed above were converted to a 7-point rating scale (point 7 is the highest score) [1]. One of the values from 1 to 7, depending on which interval it fell into, was assigned to each value of the indicator. The interval was calculated as follows:

$$(min + i * step; min + (i + 1) * step], \quad (4)$$

where $step = (max - min)/7$, max, min are the maximum and minimum values of the indicator, respectively, $i = 0, \dots 6$. If a larger value was considered the best for an indicator, then the highest score 7 was assigned to the values from the last interval $(min + 6 * step; max]$. And vice versa, if a lower value was the best, then the highest score 7 was assigned to the values from the first interval $[min, \min + step]$.

The score 0 was given if the value of the property was not acceptable at all and could not qualify for use in the security system. So, the score 0 was assigned to methods if there were no landmarks. The SCRFD-500MF method was assigned MaxFaceAngle=0, as it has confidence above 90. DSFD was assigned DetectionTime=0, as it could not be applied in real time.

The result of the rating scale conversion is shown in Fig. 13.

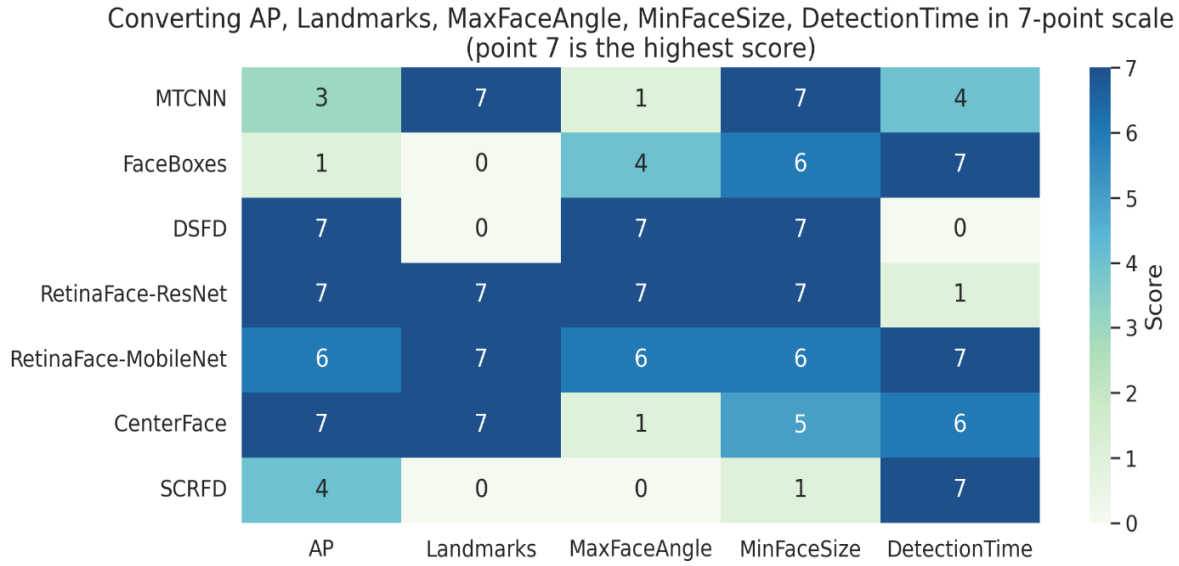

Figure 13: Converting the indicator values to a 7-point rating scale

When choosing a detection method, the main requirement is time costs. RetinaFace-MobileNet0.25, FaceBoxes, and SCRFD-500MF have a score of 7 in the DetectionTime indicator, while CenterFace has a score of 6. From the considered models, only RetinaFace-MobileNet0.25 and CenterFace have landmarks. CenterFace has the best AR score, which is 7, against 6 for RetinaFace-MobileNet0.25. But the AR was measured on a very difficult WIDER FACE (Hard) dataset. In addition, the RetinaFace-

MobileNet0.25 model shows a significantly higher rating for MaxFaceAngle, which is 6, against 1 for CenterFace. In addition, RetinaFace-MobileNet0.25 has 1 more score for MinFaceSize. Summing up the scores for each method for all properties, we found that the RetinaFace-MobileNet0.25 model has the highest score of 32 out of 35 possible.

After analyzing all the information, it was decided to use the RetinaFace-MobileNet0.25 method for the security system at the first stage.

4. Using Selected Detection Method in Enterprise Security System

The purpose of this section is to show the practical results and feasibility of using the selected RetinaFace-MobileNet0.25 detection method in real-world conditions, namely in the video surveillance-based security system where faces were detected from the real video stream, as opposed to the artificial images used in the research.

A security system based on face recognition from surveillance cameras includes the following stages: face detection, face normalization, face feature vector acquisition (embedding), and classification. Since the paper is devoted to a detailed study of the detection stage, the focus is on experimenting with detection methods and choosing the best one, the other stages are described very briefly. As mentioned above, the RetinaFace-MobileNet0.25 method was chosen for detection. Normalization was performed by rotating the face by an angle calculated from landmarks. The embeddings of the face were calculated and converted into a vector description using ResNet, which returned a 256-point face embedding. The Support Vector Machine (SVM) method was used as a recognition method.

First, the system had to be trained using SVM. During the training phase, the system administrator creates profiles of workers (classes) that will be used for recognition. The administrator added photos of the corresponding person to each created profile, which should contain an image of only one person, with the face occupying 30%. Additionally, photos could be added from the cameras when the person stood in front of the camera and followed the administrator's commands. After filling in the user profiles, the administrator initiated SVM training.

C++, OpenCV, JavaScript, and React were used to create the recognition and training system.

The following microservices were allocated to create the system: a face recognition system that detects faces, records data in a database, extracts facial features, normalizes them, and classifies them; a learning system that trains a classification model; front-end – user interface; back-end – the layer between the recognition system, training system, database, and user interface.

An example of security system operation is shown in Fig. 14. The system contains confidential data, which is hidden in the figure, and the photos are blurred.

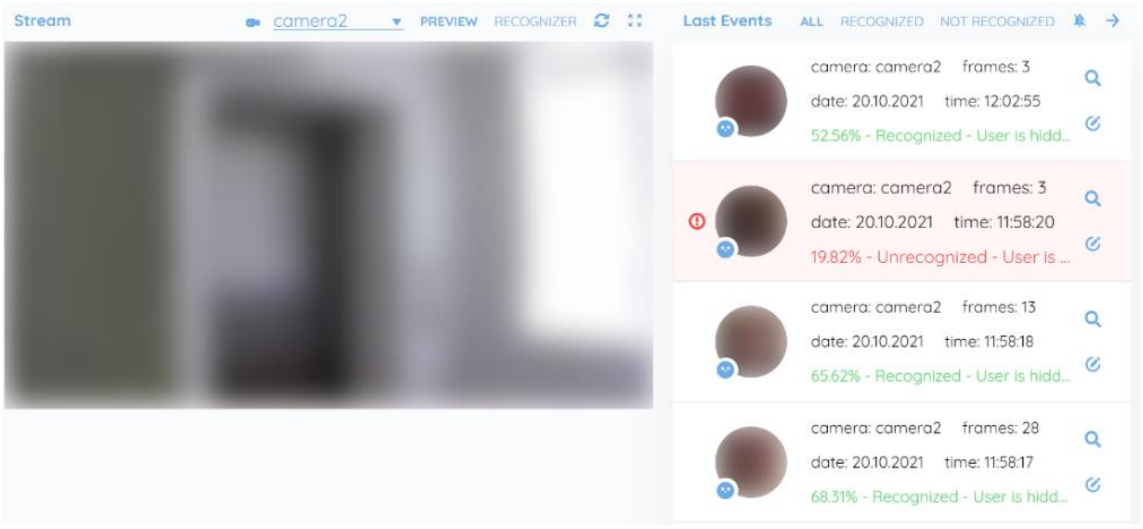

Figure 14: Example of security system operation

A data collection function was created to analyze the system's performance. It allows adding data to the dataset from cameras or manually, the administrator selects the data and assigns it to the appropriate class. This feature is presented in the user interface, which allows users to easily and quickly add or remove data and test the system. The dataset contains more than 200 classes and about 17,000 photos. On average, 100 image examples have been collected for each class. The system also allows users to exclude data from testing selectively. This feature adds more convenience to testing when, for example, an administrator needs to calculate the recognition accuracy of one class only.

The following classical metrics were chosen to assess the quality of the video surveillance-based security system [27]:

- True Positive Rate (tpr) or Recall – the proportion of samples identified correctly among the total number of class objects submitted for recognition; in the case of the system under study, this corresponds to the percentage of cases of correct human recognition among the total number of experiments for this person;
- Positive Predictive Value (ppv) or Precision – the proportion of correctly identified samples to all objects that were assigned to this class in the classification process; in the case of the system under consideration, this corresponds to determining the percentage of cases where a person is recognized correctly among the total number of faces assigned to this class.
- f1-score is the harmonic mean between precision and recall;
- tpr IQR – interquartile range (IQR) of tpr values at all classifier thresholds;
- f1 IQR – interquartile range (IQR) of f1 values at all classifier thresholds;
- median tpr, median ppv, median tpr, median f1, mean f1 – generalized metrics for evaluating the system as a whole, calculated on the basis of tpr, ppv, f1 of each class.

For different thresholds of the model, the corresponding metrics take on different values, so by changing the threshold, we can change the result of the system operation. The ideal for the system would be if the sensitivity values of ppv and tpr were equal to 1. But it is only possible for a model that classifies faultless. By their meaning, ppv and tpr are antagonistic metrics (as ppv increases, tpr decreases), so the f1-measure is used to take both of them into account.

For the studied system, this corresponds to the contradiction between security (refusal to let in a person who is not entitled to do so) and usability (reducing the number of denials for a person who is allowed to enter).

Since many classes are defined in the model, the one-vs-rest method was chosen to calculate the metrics for each class, and then the arithmetic mean of the metric (macro-average) was calculated. For more information, the median values and interquartile ranges of the metrics were also calculated (Fig. 15). To select the optimal model, we chose threshold=0.4, which corresponds to the highest value of f1-score.

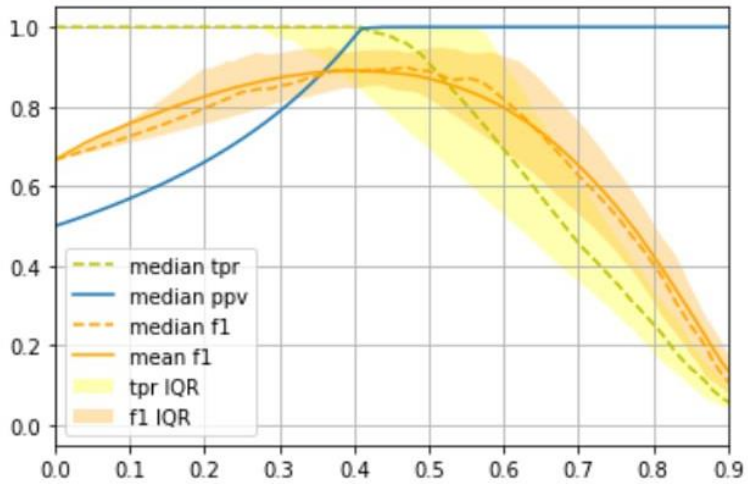

Figure 15: Regularities of change in the values of the metrics median tpr, median ppv, median tpr, median f1, mean f1, tpr IQR and f1 IQR from the threshold value

To evaluate the quality of the developed system, which would not depend on the threshold value, we calculated the AUC-ROC value – the area under the error curve (Receiver Operating Characteristic curve), and its value was 0.999997. To sum up, the developed system using the RetinaFace-MobileNet0.25 detection model at the first stage meets the requirements for the enterprise security system specified in Section 2.

5. Discussions

In this paper, we use AP values for the models from the works [11–16,18] MTCNN, FaceBoxes, DSFD, RetinaFaceResNet125, RetinaFaceMobileNet0.25, CenterFace, SCRFD-500MF, which were measured on the WIDER FACE (Hard) validation dataset [19]. According to this accuracy, the methods can be ranked as follows: RetinaFaceResNet125> DSFD> CenterFace> RetinaFaceMobileNet0.25> SCRFD-500MF> MTCNN> FaceBoxes. At the beginning of the research, it was assumed that in the experiments on the robustness of methods in the conditions of rotation and resizing of faces, the methods would be ranked in a similar order. However, the model rankings obtained from the experiments differed significantly. The use of synthetic datasets for experiments can explain this difference. The synthetic datasets were needed to control the values of the face rotation and image size parameters. However, in further work, it is desirable to investigate the accuracy of the fastest detection models, namely RetinaFaceMobileNet0.25, CenterFace, SCRFD-500MF, by AP indicator on working real datasets with which the security system will operate.

6. Conclusions

The scientific novelty of this work is the further development of face recognition methods, where the first stage is object detection.

The practical significance of the obtained results is a thorough study of the MTCNN, FaceBoxes, DSFD, RetinaFaceResNet125, RetinaFaceMobileNet0.25, CenterFace, SCRFD-500MF detection models to further use the best one for the enterprise security system based on face recognition from the video stream of surveillance cameras. Particular attention was paid to the compromise between the speed and accuracy of the studied methods.

Experiments were conducted on the robustness of the models to face rotation in different planes and face resizing, and time costs were also evaluated. The experiments were conducted on pre-prepared datasets. As a result of the research, it can be noted that:

- The best models in terms of rotation were RetinaFaceResNet125, DSFD, RetinaFaceMobileNet0.25, which confidently (confidence ≥ 0.9) detect faces with rotations in the range [-45;45], which is a sufficient requirement for use in a security system. The MTCNN, FaceBoxes, CenterFace, and SCRFD-500MF models also work with rotated faces but have lower confidence and a smaller range of face angles.
- In experiments with different face sizes, the best results were shown by MTCNN, DSFD, RetinaFace-ResNet, FaceBoxes, RetinaFace-MobileNet0.25 models, which detect images starting at 75×75 px with a confidence of ≥ 0.9. The CenterFace and SCRFD0.5GF models have significantly lower confidence but detect very small faces in some cases.
- Detection time measurements showed that the fastest models are RetinaFace-MobileNet0.25 and FaceBoxes, which spend less than 47 ms to process one card for VGA images and 100 ms for HD size. The next rank belongs to the SCRFD-500MF, CenterFace. For the MTCNN, and RetinaFace-ResNet125 models, detection time exceeds 100 ms even for VGA images. The DSFD model cannot be applied in real-time, even for VGA images.

To finally select the best model for further use in the security system, the quantitative values of their properties were converted to a 7-point scale. The following properties were considered: the claimed accuracy of AP methods in primary publications [11–16,18], the presence of landmarks, the maximum range of rotation angle, the minimum face size detected with a confidence of more than 0.9, the average frame processing time of one frame for VGA images (Table 4). After analyzing the ratings, the

RetinaFace-MobileNet0.25 model was chosen as the best for use in the security system, as it is one of the fastest, has landmarks, and is robust to rotation and changes in face size.

The RetinaFace-MobileNet0.25 was used in the enterprise security system based on face recognition from surveillance cameras. as the first stage detection model. Then there were the stages of face normalization, obtaining embeddings, and classification. ResNet was used for embedding, and SVM was used for classification (recognition). The accuracy of the system was 0.999997 according to the AUC-ROC value, which shows the feasibility of using RetinaFace-MobileNet0.25 in the system to solve the face detection problem.

In the future, it is advisable to investigate other stages more thoroughly, especially the matter of the maximum number of classes for which SVM can be used as a classifier, the use of other classification methods, and the specifics of retraining the security system.

7. Acknowledgements

The authors express their gratitude to SYTOSS s.r.o. Bratislava, the Slovak republic, represented by the director Oleksiy Matikaynen for providing equipment for research and employees for participation in the experiments for security system configuration.

The work is funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under project No. 09I03-03-V01-00115.

8. References

[1] O. Yakovleva, K. Nikolaieva, Research of Descriptor Based Image Normalization and Comparative Analysis of Surf, SIFT, Brisk, Orb, Kaze, Akaze Descriptors, Advanced Information Systems 4 (4) (2020) 89101. doi:10.20998/2522-9052.2020.4.13.

[2] A. Kovtunenko, O. Yakovleva, Doslidzhennia sumisnoho vykorystannia matematychnoi morfolohii ta zhortkovikh neironnykh merezh dla virishennia zadachi rozpiznavannia tsikavnykiv, Visnyk Natsionalnoho tekhnichnoho universytetu "KhPI". Seriia: Systemnyi analis, upravlinnia ta informatsiini tekhnolohii, 1 (3) (2020) 24–31. doi:10.20998/2079-0023.2020.01.05.

[3] V. Gorokhovatskyi, I. Tvoroshenko, Image Classification Based on the Kohonen Network and the Data Space Modification, in: CEUR Workshop Proceedings: Computer Modeling and Intelligent Systems (CMIS-2020), 2020, pp. 1013–1026. doi:10.32782/cmis/2608-76.

[4] Y. Daradkeh, V. Gorokhovatskyi, I. Tvoroshenko, S. Gadetska, M. Al-Dhaifallah, Methods of classification of images on the basis of the values of statistical distributions for the composition of structural description components, IEEE Access 9 (2021) 92964–92973. doi:10.1109/ACCESS.2021.3093457.

[5] P. Viola, M. Jones, Rapid Object Detection Using a Boosted Cascade of Simple Features, in: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), 2001. doi:10.1109/cvpr.2001.990517.

[6] N. Dalal, B. Triggs, Histograms of Oriented Gradients for Human Detection, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005. doi:10.1109/cvpr.2005.177.

[7] H. Li, Z. Lin, X. Shen, J. Brandt, G. Hua, A convolutional neural network Cascade for face detection, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 5325–5334. doi:10.1109/cvpr.2015.7299170.

[8] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587. doi :10.1109/cvpr.2014.81.

[9] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, pp. 1137–1149. doi : 10.1109/tpami.2016.2577031.

[10] L. Chengjun, H. Wechsler, Gabor feature based classification using the Enhanced Fisher linear discriminant model for face recognition, in: IEEE Transactions on Image Processing, 2002, pp. 467–476. doi:10.1109/tip.2002.999679.

[11] K. Zhang, Z. Zhang, Z. Li, Y. Qiao, Joint face detection and alignment using multitask cascaded convolutional networks, IEEE Signal Processing Letters 23(10) (2016) 1499–1503. doi:10.1109/lsp.2016.2603342.
[12] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, S. Z. Li, FaceBoxes: A CPU real-time face detector with high accuracy, in: 2017 IEEE International Joint Conference on Biometrics (IJCB), 2017, pp. 1–9. doi:10.1109/btas.2017.8272675.
[13] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 39(6) (2016) 1137–1149. doi:10.1109/tpami.2016.2577031.
[14] J. Deng, J. Guo, E. Ververas, I. Kotsia, S. Zafeiriou, Retinaface: Single-shot multi-level face localisation in the wild, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 5203–5212. doi:10.1109/cvpr42600.2020.00525.
[15] Y. Xu, W. Yan, G. Yang, J. Luo, T. Li, J. He, Centerface: Joint face detection and alignment using face as point, Scientific Programming (2020) 1–8. doi:10.1155/2020/7845384.
[16] J. Guo, J. Deng, A. Lattas, S. Zafeiriou, Sample and computation redistribution for efficient face detection, 2021. URL: https://arxiv.org/abs/2105.04714.
[17] E. Zhang, Y. Zhang, Average precision, Encyclopedia of Database Systems (2009) 192–193. doi:10.1007/978-0-387-39940-9_482.
[18] InsightFace Model Zoo, 2021. URL: https://github.com/deepinsight/insightface/tree/master/model_zoo.
[19] S. Yang, P. Luo, C. C. Loy, X. Tang, Wider face: A face detection benchmark, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 5525–5533. doi:10.1109/cvpr.2016.596.
[20] Pretrained Pytorch Face Detection (MTCNN) and facial recognition (InceptionResnet) models, 2019. URL: https://github.com/timesler/facenet-pytorch.
[21] A pytorch implementation of faceboxes, 2017. URL: https://github.com/zisianw/FaceBoxes.PyTorch.
[22] A high-performance pytorch implementation of face detection models, including RetinaFace and DSFD. 2019. URL: https://github.com/hukkelas/DSFD-Pytorch-Inference.
[23] Star-Clouds/Centerface: Face detection, 2019. URL: https://github.com/Star-Clouds/CenterFace.
[24] InsightFace Python Library, 2022. URL: https://github.com/deepinsight/insightface/tree/master/python-package.
[25] Unique, worry-free model photos. Generated Photos, 2022. URL: https://generated.photos.
[26] Home of the blender project – free and open 3D creation softwareб 2022. URL: https://www.blender.org.
[27] A. González-Ramírez, J. Lopez, D. Torres, I. Yañez-Vargas, Analysis of multi-class classification performance metrics for remote sensing imagery imbalanced datasets, Journal of Quantitative and Statistical Analysis (2021) 11–17. doi:10.35429/jqsa.2021.22.8.11.17.