

Article

The Study of Mathematical Models and Algorithms for Face Recognition in Images Using Python in Proctoring System

Ardak Nurpeisova ^{1,*}, Anargul Shaushenova ^{1,†}, Zhazira Mutalova ^{2,†}, Zhandos Zulpykhar ^{3,†},
Maral Ongarbayeva ^{4,†}, Shakizada Niyazbekova ^{5,6,†}, Alexander Semenov ^{7,†} and Leila Maisigova ^{8,†}

- ¹ Department of Information Systems, Faculty of Computer Systems and Professional Education, S. Seifullin Kazakh Agro Technical University, Nur-Sultan 010000, Kazakhstan
- ² Institute of Economics, Information Technologies and Professional Education, Higher School of Information Technologies, Zhangir Khan West Kazakhstan Agrarian Technical University, Uralsk 090000, Kazakhstan
- ³ Department of Computer Science, Faculty Information Technology, L.N. Gumilyov Eurasian National University, Nur-Sultan 010000, Kazakhstan
- ⁴ Department of Information-Communication Technology, Faculty of Natural Sciences, International Taraz Innovative Institute, Taraz 080000, Kazakhstan
- ⁵ Department of Banking and Monetary Regulation, Financial University under the Government of the Russian Federation, Moscow 125993, Russia
- ⁶ Research and Education Center “Sustainable Development”, Moscow Witte University, Moscow 115432, Russia
- ⁷ Administration, Moscow Witte University, Moscow 115432, Russia
- ⁸ Department of Accounting, Analysis and Audit, Faculty of Economics, Ingush State University, Magas 386001, Russia
- * Correspondence: nurpeisova.ardak81@gmail.com; Tel.: +7-747-313-2696
- † These authors contributed equally to this work.



Citation: Nurpeisova, A.; Shaushenova, A.; Mutalova, Z.; Zulpykhar, Z.; Ongarbayeva, M.; Niyazbekova, S.; Semenov, A.; Maisigova, L. The Study of Mathematical Models and Algorithms for Face Recognition in Images Using Python in Proctoring System. *Computation* **2022**, *10*, 136. <https://doi.org/10.3390/computation10080136>

Academic Editors: Christos Troussas, Cleo Sgouropoulou, Akrivi Krouska, Ioannis Voyiatzis and Athanasios Voulodimos

Received: 7 July 2022

Accepted: 28 July 2022

Published: 9 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



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

Abstract: The article analyzes the possibility and rationality of using proctoring technology in remote monitoring of the progress of university students as a tool for identifying a student. Proctoring technology includes face recognition technology. Face recognition belongs to the field of artificial intelligence and biometric recognition. It is a very successful application of image analysis and understanding. To implement the task of determining a person's face in a video stream, the Python programming language was used with the OpenCV code. Mathematical models of face recognition are also described. These mathematical models are processed during data generation, face analysis and image classification. We considered methods that allow the processes of data generation, image analysis and image classification. We have presented algorithms for solving computer vision problems. We placed 400 photographs of 40 students on the base. The photographs were taken at different angles and used different lighting conditions; there were also interferences such as the presence of a beard, mustache, glasses, hats, etc. When analyzing certain cases of errors, it can be concluded that accuracy decreases primarily due to images with noise and poor lighting quality.

Keywords: proctoring systems; AI-based AEPS (artificial intelligence-based automated exam proctoring systems); algorithm; artificial intelligence; mathematical model; person detection

1. Introduction

The COVID-19 outbreak and its associated restrictions have created huge challenges for schools and higher education institutions, requiring urgent action to maintain the quality of teaching and student assessment.

The development of information technology, online learning and online exams are becoming more widespread. Distance learning and distance exams have facilitated the work of teachers and students to a certain extent, and at the same time, have balanced the gap in educational resources between regions. However, there are certain disadvantages in distance exams: the absence of observers makes it easy to cheat on the distance exam, which affects the fairness of the exam and the quality of education.

The use of information technology and the introduction of distance education in higher education requires the implementation of effective actions to control students. Proctoring performs the functions of recognition, tracking and evaluation during the execution of control, milestone and final work. This process can be live, automated and semi-automated, depending on the role of man and machines in it. Proctoring in higher education should be based on the theses of adaptation, standardization, information security, personalization and interactivity.

A number of educational institutions believe that proctoring technology is necessary to prevent fraud. A number of other educational institutions and students are concerned about the difficulties associated with this approach. Automated proctoring programs provide examiners with the tools to prevent fraud. These programs can collect system information, block network access, and parse keystrokes. They may also use computer cameras and microphones to record students and their surroundings.

Since the teacher does not have the opportunity to often memorize students, it becomes necessary to determine the examinees, as well as the normative support for the process of passing the exam. In the proctoring system, the trajectory detection and tracking technology based on face recognition uses the regional feature analysis algorithm, which integrates computer image processing technology and the principle of biostatistics. Mathematical model building has broad prospects for development.

Facial recognition falls under the category of artificial intelligence and uses computer optics, acoustics, physical sensors, biological statistical principles and advanced mathematical methods to create models that turn human physiological characteristics into identifiable ones. Face recognition can be widely used in many scenarios, such as airports, scenic spots, hotels, railway stations and other places.

With the rapid development of artificial intelligence technologies, facial recognition has become widely used in social work and life, while technologies such as face checking and door opening provide convenience, their own security issues have often led to the exposure of user security vulnerabilities, leading practitioners to realize the need to improve the facial recognition system. Face identification is a difficult problem in the field of image review and computer vision. Information security becomes a very significant and difficult task [1–8].

Most Kazakhstan's universities have adapted to the situation with distance learning and are ready to organize intermediate and final attestation of students online. Evaluation of acquired knowledge is the most difficult stage of distance learning to implement. In the development of an automated proctoring system, an important role is played by the face recognition system. With a face recognition system, you can eliminate the influence of the human factor in proctoring by controlling all students equally. We explore the current topic of face recognition and our goal is to study the model of face recognition. We also apply the created face recognition system at our university.

2. Materials and Methods

The potential benefits of safeguards such as remote exam fraud research and remote verification and the methods used in education have been discussed in the works of many authors [9–14]. These articles raise current questions regarding cheating and other inappropriate test-taker behavior, how to deal with such behavior and whether remote proctoring provides an effective solution. Research in the field of pattern recognition has been confirmed by numerous works of scientists abroad [15–17]. The specification of the direction by means and technologies of proctoring programs was published in the works of Kazakh researchers [18–20].

A large number of works are devoted to the mathematical formulation of the problem of automatic pattern recognition. Explicitly or implicitly, questions of defining the initial concepts are intertwined in the circle of these questions. When discussing general issues of image recognition, geometric representations are widely used. Although it is clear that any kind of multidimensional, and even more so, infinite-dimensional constructions, have only

an auxiliary character of explanation, the nature of methodological means are not intended for actual schematic implementation. The review will also use a geometric interpretation of the basic identification facts. A huge number of works are devoted to the mathematical formulation of the problem of automatic pattern recognition. When discussing general issues of pattern recognition, geometric representations are widely used [21–25].

Formulation of the problem.

Within the framework of this work, the task is to study the parameters of face recognition algorithms. The research process can be divided into several tasks, namely:

- (1) review methods and algorithms for face detection and recognition;
- (2) describe mathematical models of face recognition;
- (3) implementation of the face recognition algorithm;
- (4) development of recommendations for improving the algorithm to obtain a given accuracy.

To test the influence of parameters on the accuracy of the results of the face recognition algorithms, we chose the Python programming language. The Python language supports the basic programming paradigms needed to get the job completed. Easy to manage with codes, a huge number of useful libraries (NumPy library, OpenCV, Dlib, OpenFace) were used. We used the Viola–Jones algorithm for face detection.

On the basis of proctoring, we used 400 pictures of students and 40 pictures of each student. Currently, there are different methods of face recognition. Here, we have used the libraries from OpenCv. To create a face detection instance model, we used `cv2.face_recognition` function of OpenCV API, then used the `face_detector.py` function.

We used photographs taken from different angles, and in the experiment, we used 400 photographs of 40 students. Students were asked to take pictures with different facial expressions. The Face Detection operation supports images that meet the following requirements: jpeg, png, gif (first frame) or bmp format; the file size is between 1 KB and 6 MB and the image size ranges from 36×36 pixels to 4096×4096 pixels. The following factors significantly influence the probability of correct image recognition: resolution (size)—the most stable (critical resolution) between the best resolution and the smallest resolution; brightness; lighting; grip angle.

There are various methods that allow you to determine the features of a person from a face image. The main criteria for evaluating methods are the computational difficulty of the algorithms and the probability of correct recognition. The choice of the pattern recognition method depends on the nature of the problem. Principal Component Analysis, Independent Component Analysis, Active Shape Model and Hidden Markov Model are some of the most important dimensionality reduction or early face detection algorithms. These methods have a wide range of applications in data compression to remove redundancy and eliminate noise in data. The advantages of these methods are that if emotions, lighting, etc. are present in the images, then additional components will appear; thus, it will be easy to save and search for images in large databases, and recreate images [26–28]. The main difficulty of the methods is the high demand on images. Images must be obtained in low light conditions, at one angle, and high-quality pre-processing of images must be carried out, leading to standard conditions.

Support Vector Machines is a supervised learning method that can be widely used in statistical classification and regression analysis. Support vector machines are generalized linear classifiers. The characteristic of this family of classifiers is that they can minimize the empirical error and maximize the geometric edge area at the same time, which is why the support vector machine is also called the maximum edge area classifier. Neural Networks is a model for a specific classification task; the focus is on “learning” and the basis of machine learning is thinking. Neurons can work independently and process the information received, that is, the system can process input information in parallel, possessing the ability to self-organize and self-learn. The downside of Neural Networks is that the sample set has a lot of impact and requires a lot of computing power [29–31]. A large amount of redundant data are generated, leading to low training efficiency, and a large number of false positive samples can appear in the classification. Different face

recognition methods have different probabilities of correct recognition, and these methods are interdependent. It depends on the parameter of the recognized object (Table 1).

Table 1. Literary Review.

Face Recognition Methods	Number of Images	Correct Recognition Result
Principal Component Analysis [23]	400	79.65%
Principal Component Analysis, Component Analysis [23]	400	92.34%
Independent Component Analysis [15–17]	100	81.3%
Wavelet Transform [13]	100	80–90%
Active Shape Model [25]	100	78.12–91%
Hidden Markov Model [15]	200	84%
Eigenfaces Method [23]	70	92–95%
Support Vector Machines [25]	200	85–91%
Neural Networks [32]	200	93.7%

Facial recognition is considered as a biometric authentication procedure in all automatic individual authentication systems [33–35]. Many organizations and governments rely on this method to secure public places such as airports, bus stops, train stations, etc. Most current face recognition models require highly accurate machine learning to recognize labeled face datasets [36,37]. The most advanced face recognition models such as Facenet [38–40] have shown recognition accuracy of 99% or better. Comprehensive experiments conducted with the Georgia Tech Face Dataset, the head pose image dataset and the Robotics Lab Face dataset showed that the proposed approach is superior to other modern mask recognition methods. As published in other review articles, invariant methods of face recognition [41] or illumination [42], dynamic face recognition from an image [43–47], multimodal face recognition using 3D and infrared modalities [48–50], attack detection methods (anti-counterfeiting) [51].

Convolutional Neural Networks (CNNs) are the most commonly used type of deep learning method for face recognition. The main advantage of the deep learning method is that you can use a large amount of data for training to get a reliable idea of the changes that occur in the training data. This method does not require the development of specific characteristics that are robust to various types of class differences (such as lighting, posture, facial expression, age, etc.) but can be extracted from the training data. The main disadvantage of deep learning methods is that they need to use very large datasets to train, and these datasets must contain enough changes to be able to generalize to patterns that have never been seen before. Some large-scale face datasets containing images of natural faces have been made public and can be used to train CNN models. In addition to learning recognition features, neural networks can also reduce dimensionality and can be trained with classifiers or use metric learning methods. CNN is considered an end-to-end learning system and does not need to be combined with any other specific methods [52–54].

A pattern recognition computer system basically consists of three interrelated but clearly differentiated processes; namely, data generation, pattern analysis and pattern classification. Data generation consists of converting the input template's original information into a vector and giving it a form that is easy for a computer to process. Pattern analysis is the processing of data, including feature selection, feature extraction, data compression by size, and determination of possible categories. Image classification is the use of information obtained from image analysis to teach a computer to formulate recognition criteria in order to classify recognition images.

Face recognition system includes the following blocks (Figure 1):

- a block for constructing an object recognition model (search for the coordinates of a person’s surface in a figure, determining the information zonal location, preprocessing and normalization;
- object recognition authentication block (authentication algorithms for an object to be identified by controlling access to the photo recognition system of a user registered in the database);
- a block for calculating information identification marks (convolutional neural networks, correlation indicators, Minkowski distance, etc.).

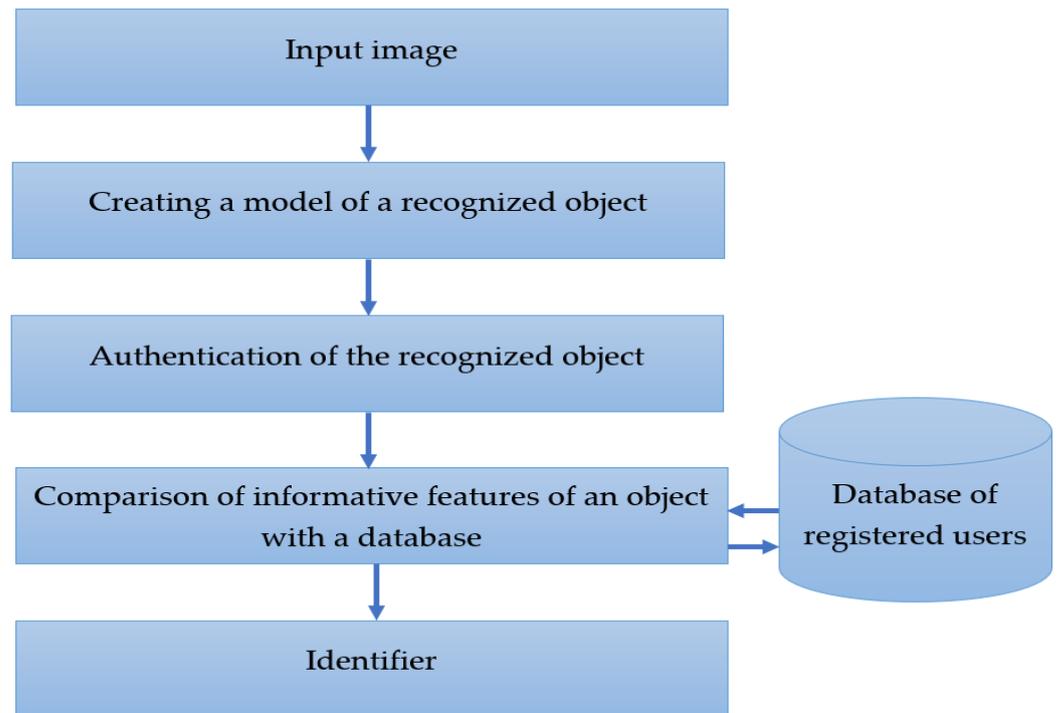


Figure 1. Structure of the face image recognition system.

The mathematical model of face recognition is presented below. Let us define a set of face images in the database $X_1, X_2, X_3, \dots, X_L$. The sets are divided into L classes, where each class corresponds to a registered person [55–58]. For each image, we define a vector of K values:

$$\mu = (\mu_1, \mu_2, \mu_3, \dots, \mu_k)^T \tag{1}$$

where τ is the transpose operator. For each image (1), we define the distance function. Distance functions $d(\mu_l, \mu_s)$ for the feature vector at the greatest distance μ in the input form belongs to the class X_L ,

$$d(\mu_l, \mu_s) > d(\mu_j, \mu_s), l \neq j, l, j = 0, 1, \dots, L - 1 \tag{2}$$

For the distance function (2), the class X_L must exceed a pre-computed threshold value $d(\mu_l, \mu_s) > \tau_c$. The input to the face recognition algorithm is an image, and the output is a sequence of face frame coordinates (0 face frames or 1 face frame or multiple face frames). Typically, the output frame of face coordinates is an upward-facing square, but there are also some face detection technologies that output an upward-facing rectangle or a rectangle with a rotation direction. The conventional face detection algorithm is basically a “scanning” and “distinguishing” process, that is, the algorithm scans a range of images and then determines in turn whether a candidate area is a face. Therefore, the calculation speed of the face detection algorithm is related to the image size and image content. The input to the face registration algorithm is a “face image” plus a “frame of face coordinates”, and the output is a sequence of coordinates of the key points of the facial features. The number of facial

feature keypoints is a predetermined fixed value that can be defined according to different semantics (typically 5 points, 68 points, 90 points, etc.).

Using a mathematical model to determine the integral image, you can find the coordinates of the face (Viola and Jones algorithm):

$$I_i(x, y) = \sum_{i=1}^{x \cdot y} i(x', y') \tag{3}$$

where $I_i(x, y)$ —the value of the i -th element of the integral image with coordinates is the (x, y) , (x', y') —brightness of the pixel of the image under consideration with coordinates (x', y') . The integral image (3) is calculated regardless of the size or location of the image and is used to quickly calculate the brightness of given parts of the image. Sign s_i —the sum of the brightness of the pixels lying in the white areas is subtracted from the sum of the intensities of the pixels lying in the black areas:

$$S_i = \sum_{i=0}^{y \cdot x} r_{i,k} - \sum_{i=0}^{y \cdot x} r_{i,f} \tag{4}$$

where $\sum_{i=0}^{y \cdot x} r_{i,k}, \sum_{i=0}^{y \cdot x} r_{i,f}$ is the sum of the brightness of the pixels.

The Adaboost Method, directly determining the error rate of a simple classifier, avoids time-consuming processes such as iterative learning and statistical probability distribution. The closer (4) is to 0, the lower the brightness; the closer the value is to 255, the higher the brightness. There are n training samples $(x_1, y_2), (x_2, y_2), \dots, (x_n, y_n)$ where $x_n \in X, y_n \in Y$ corresponds to negative and positive sample samples. In the training set c , there are m samples of negative cases and l samples of positive cases. At the same time, there is a set of weights $w_i, 1 \leq i \leq n$, corresponding to each sample.

The expression for this classifier with a threshold value τ_c is:

$$W = \begin{cases} 1, & f_i \geq \tau_c; \\ -1, & f_i < \tau_c; \end{cases} \tag{5}$$

The “best” strong classifier is calculated from a fixed number of weak classifiers. The expression for a strong classifier is shown:

$$W = \begin{cases} 1, & \sum_{c=1}^C a_c w_c \geq \frac{1}{2} \sum_{c=1}^C a_c; \\ 0, & \sum_{c=1}^C a_c w_c < \frac{1}{2} \sum_{c=1}^C a_c; \end{cases} \tag{6}$$

$$a_c = \log \frac{1}{\beta_c} \tag{7}$$

where w_c is weak classifier; a_c, β_c are the weight coefficients of the weak classifier; c is the number of the current weak classifier, $C = (1, \dots, C)$ is the number of weak classifiers. An iterative algorithm that implements a “strong” classifier (6) and (7), which makes it possible to achieve classifiers based on “weak” (5) compositions for learning arbitrary smaller errors. Images of the object before and after illumination can be described as follows:

$$q_j(x, y) = \frac{g_{0,j}(x, y)}{g_{1,j}(x, y) + 1} \tag{8}$$

where j is the number of the current value of the sequence $Y, g_{(b,j)}(x, y)$ —the brightness value of the pixel of the array $I_{(b,j)}$, corresponding to the image of the user’s face; b is the identifier of the pixel array, indicating the backlight mode in which the array was prepared; $b = (0, 1)$, x and y are the coordinates of the pixel in question; $x = 0, 1, \dots, W - 1$,

$y = 0, 1, \dots, H - 1$, W and H are the number of pixels corresponding to the width and height array $I_{(b,j)}$.

The measure of dispersion of all values (8) should be estimated. To describe the dispersion of the values of the numerical characteristic of the sample (9), the total characteristic of the sample variance and the mathematical expectation (10) for the mean value are introduced $q_j(x_1, y)$. To assess the degree of dispersion relative to the average value (mathematical expectation), the dispersion is calculated:

$$D(I_{0,j}, I_{1,j}) = \frac{1}{WH} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} (q_j(x, y) - q_j)^2 \quad (9)$$

where (q_j) is the mean value or mathematical expectation of a discrete random variable $q_j(x, y)$, which is calculated by the formula:

$$q_j = \frac{1}{WH} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} (q_j(x, y)). \quad (10)$$

The above presentation of the identified problems can be found in many publications devoted to user identification by face images and modeling of information systems [59,60].

3. Results

The technology for recognizing users of proctoring systems uses the methods of searching for the coordinates of a face in an image, identifying an object from a face image, an algorithm for tracking a recognized object, and detecting a substitution of a recognized object for a photo, video recording or photo mask of a registered user's face.

The advantages of OpenCV include: free open source library, fast and supported by most operating systems such as Windows, Linux and macOS. We have implemented face recognition using OpenCV and Python. To start, we used the OpenCV, dlib, face_recognition libraries and installed these libraries (Table 2).

Table 2. Preconfiguring the environment.

Preconfiguring	
Step 1:	Python 3.7 installation, Anaconda distribution.
Step 2:	OpenCV installation (we used pip to install, i.e.,: pip install opencv-python there are no version number restrictions) pip3 install opencv-python
Step 3:	Face-recognition API installation pip install dlib pip install face_recognition
Step 4:	Running the file face_detector.py
Step 5:	Running the python file face_detector.py
Step 6:	import cv2 import face_recognition
Step 7:	Let us determine the length length = int(input_movie.get(cv2.CAP_PROP_FRAME_COUNT))
Step 8:	Let us determine the file face_encoding = face_recognition.face_encodings(image)[0] known_faces = [face_encoding]
Step 9:	We ran a loop that extracted a frame from the video, found all the faces and identified them.

The following code algorithms (self, id_folder, mtcnn, sess, embeddings, images_placeholder, phase_train_placeholder, distance_treshold) check for folders with object names and determine how many images are found or defined in the folders content (Algorithm 1).

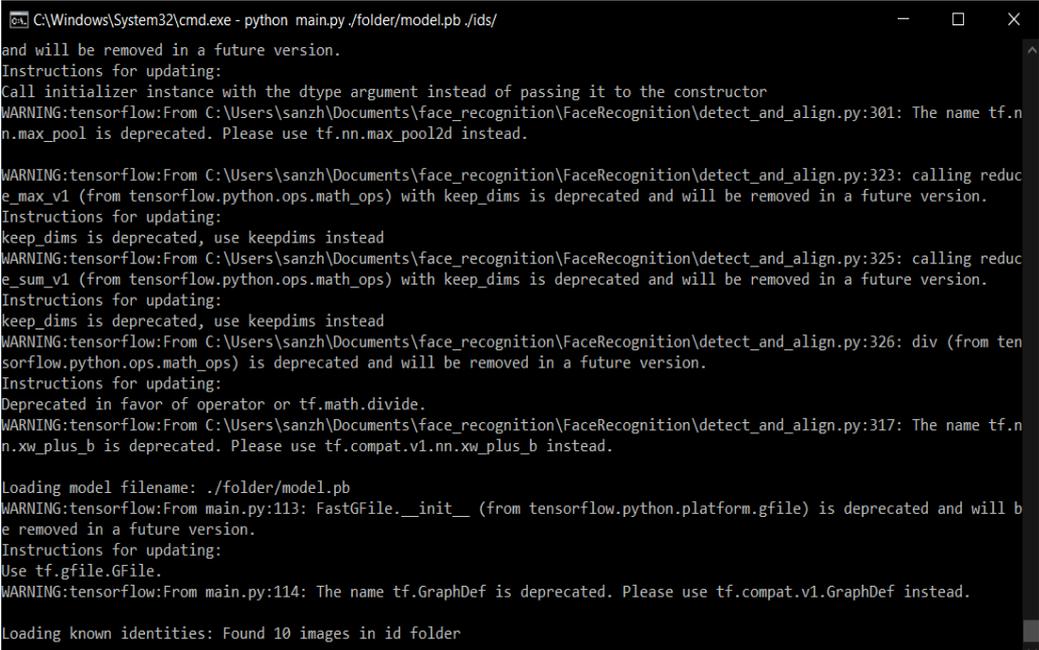
Algorithm 1 Checking the presence of folders with object names

```

1:         def init_():
2:             print("Loading known identities: ", end = "")
3:             self.distance_treshold = distance_treshold
4:             self.id_folder = id_folder
5:             self.mtcnn = mtcnn
6:             self.id_names = []
7:             self.embeddings = None
8:             image_paths = []
9:             os.makedirs(id_folder, exist_ok = True)
10:            ids = os.listdir(os.path.expanduser(id_folder))
11:            if not ids:
12:                return
13:            for id_name in ids:
14:                id_dir = os.path.join(id_folder, id_name)
15:                image_paths = image_paths + [os.path.join(id_dir, img) for img in os.listdir(id_dir)]
16:                print("Found %d images in id folder" % len(image_paths))
17:                aligned_images, id_image_paths = self.detect_id_faces(image_paths)
18:                feed_dict = images_placeholder: aligned_images, phase_train_placeholder: False
19:                self.embeddings = sess.run(embeddings, feed_dict = feed_dict)

```

Figure 2 shows the result.



```

C:\Windows\System32\cmd.exe - python main.py ./folder/model.pb ./ids/
and will be removed in a future version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it to the constructor
WARNING:tensorflow:From C:\Users\sanzh\Documents\face_recognition\FaceRecognition\detect_and_align.py:301: The name tf.nn.max_pool is deprecated. Please use tf.nn.max_pool2d instead.

WARNING:tensorflow:From C:\Users\sanzh\Documents\face_recognition\FaceRecognition\detect_and_align.py:323: calling reduce_max_v1 (from tensorflow.python.ops.math_ops) with keep_dims is deprecated and will be removed in a future version.
Instructions for updating:
keep_dims is deprecated, use keepdims instead
WARNING:tensorflow:From C:\Users\sanzh\Documents\face_recognition\FaceRecognition\detect_and_align.py:325: calling reduce_sum_v1 (from tensorflow.python.ops.math_ops) with keep_dims is deprecated and will be removed in a future version.
Instructions for updating:
keep_dims is deprecated, use keepdims instead
WARNING:tensorflow:From C:\Users\sanzh\Documents\face_recognition\FaceRecognition\detect_and_align.py:326: div (from tensorflow.python.ops.math_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Deprecated in favor of operator or tf.math.divide.
WARNING:tensorflow:From C:\Users\sanzh\Documents\face_recognition\FaceRecognition\detect_and_align.py:317: The name tf.nn.xw_plus_b is deprecated. Please use tf.compat.v1.nn.xw_plus_b instead.

Loading model filename: ./folder/model.pb
WARNING:tensorflow:From main.py:113: FastGFile.__init__ (from tensorflow.python.platform.gfile) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.gfile.GFile.
WARNING:tensorflow:From main.py:114: The name tf.GraphDef is deprecated. Please use tf.compat.v1.GraphDef instead.

Loading known identities: Found 10 images in id folder

```

Figure 2. Result of folders availability.

These lines (Algorithm 2) show the distance between the object and the camera (Algorithm 2).

Algorithm 2 Determining the distance between the object and the camera

```

1: def print_distance_table(self, id_image_paths):
2:     """Prints distances between id embeddings"""
3:     distance_matrix = pairwise_distances(self.embeddings, self.embeddings)
4:     image_names = [path.split("/")[-1] for path in id_image_paths]
5:     print("Distance matrix:::20".format(""), end = "")
6:     [print(":::20".format(name), end = "") for name in image_names]
7:     for path, distance_row in zip(image_names, distance_matrix):
8:         print(":::20".format(path), end = "")
9:         for distance in distance_row:
10:            print(":::20".format("%0.3f" % distance), end = "")
11:        print()

```

The results of Algorithm 2 (Figures 3 and 4).

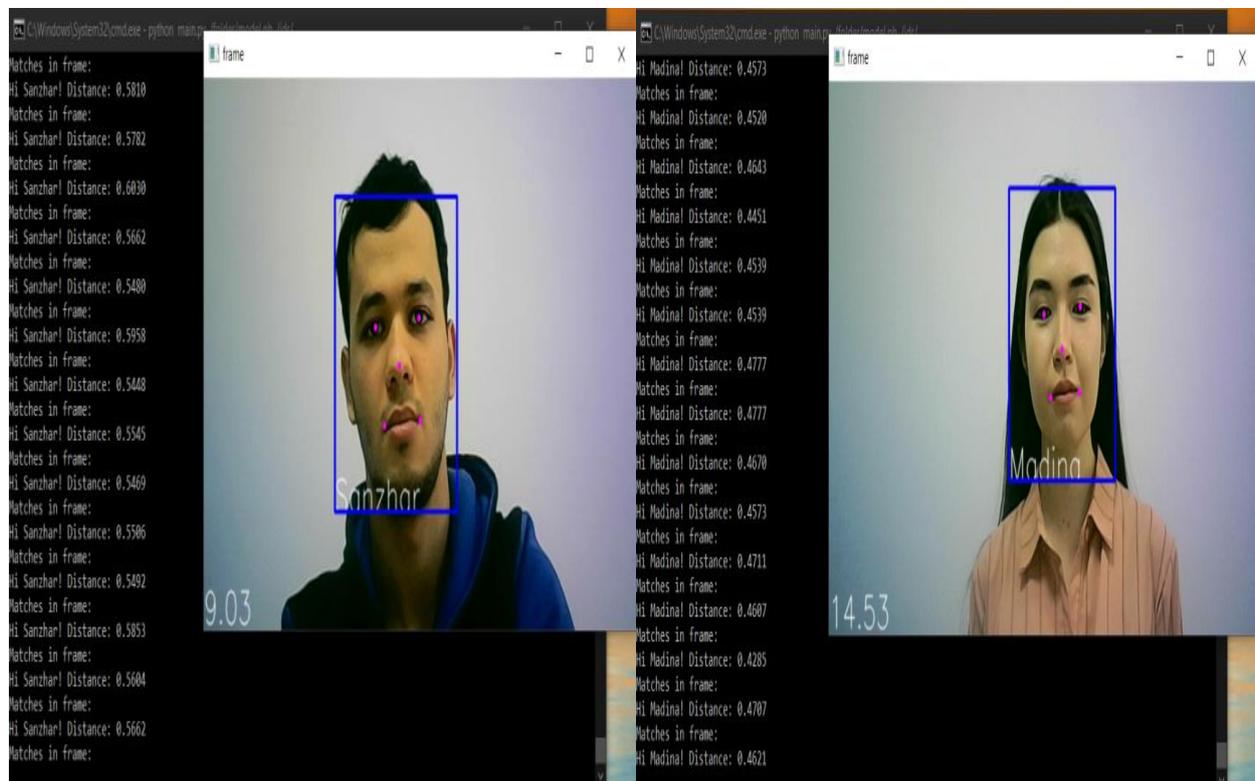


Figure 3. The result is the distance between the object and the camera.

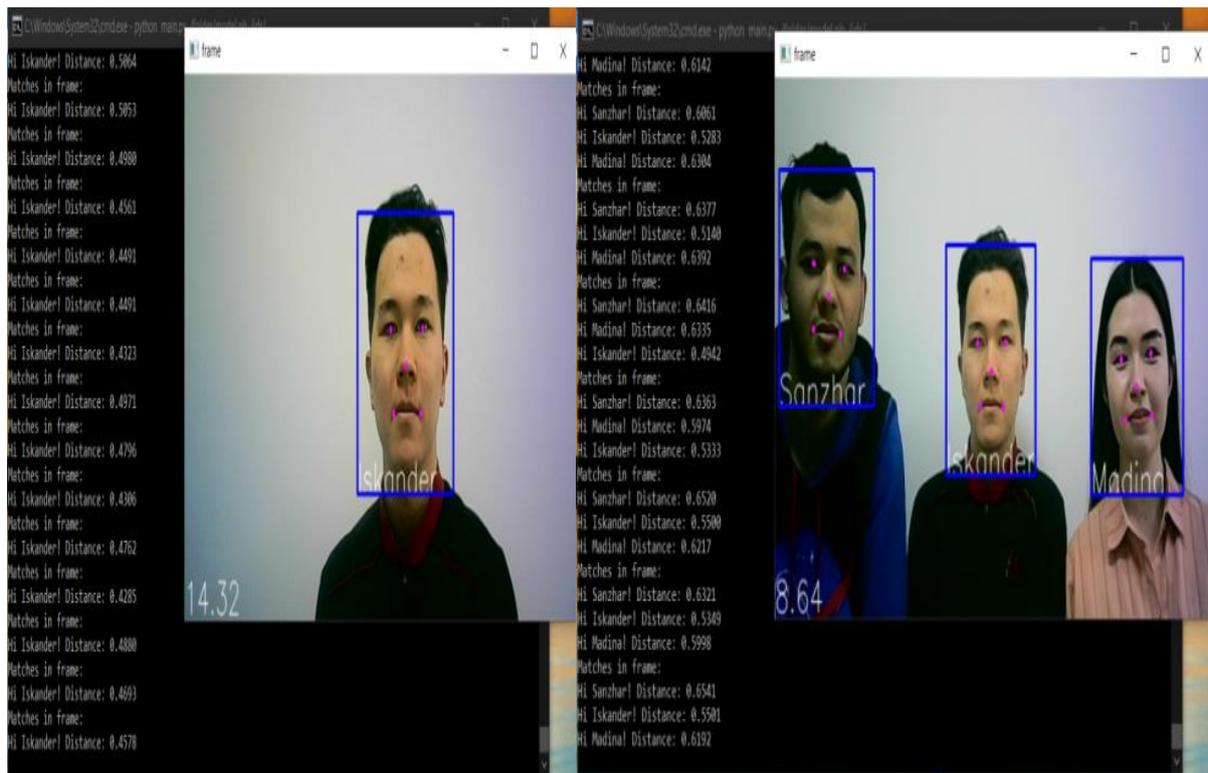


Figure 4. The result is the distance between the object and the camera.

These lines (Algorithm 3) load the neural network model that is used as the face recognition algorithm (Algorithm 3).

Algorithm 3 Neural network model for face recognition

```

1:         def load_model(model):
2:             model_exp = os.path.expanduser(model)
3:             if os.path.isfile(model_exp):
4:                 print("Loading model filename: %s" % model_exp)
5:                 with gfile.FastGFile(model_exp, "rb") as f:
6:                     graph_def = tf.GraphDef()
7:                     graph_def.ParseFromString(f.read())
8:                     graph_def.ParseFromString(f.read())
9:                     tf.import_graph_def(graph_def, name = "")
10:                else:
11:                    raise ValueError("Specify model file, not directory!")

```

These lines (Algorithm 4) allow the unknown object to be created as a new object and stored for further identification (Algorithm 4).

Algorithm 4 Creating a new object

```

1:         elif key = ord("s") and frame_detections is not None:
2:             for emb, bb in zip(frame_detections["embs"], frame_detections["bbs"]):
3:                 patch = frame_detections["frame"][bb[1] : bb[3], bb[0] : bb[2], :]
4:                 cv2.imshow("frame", patch)
5:                 cv2.waitKey(1)
6:                 new_id = easygui.enterbox("Who's in the image? Leave empty for non-valid")
7:                 if len(new_id) > 0:
8:                     id_data.add_id(emb, new_id, patch)
9:                 cap.release()
10:                cv2.destroyAllWindows()

```

Results are shown in Figure 5.

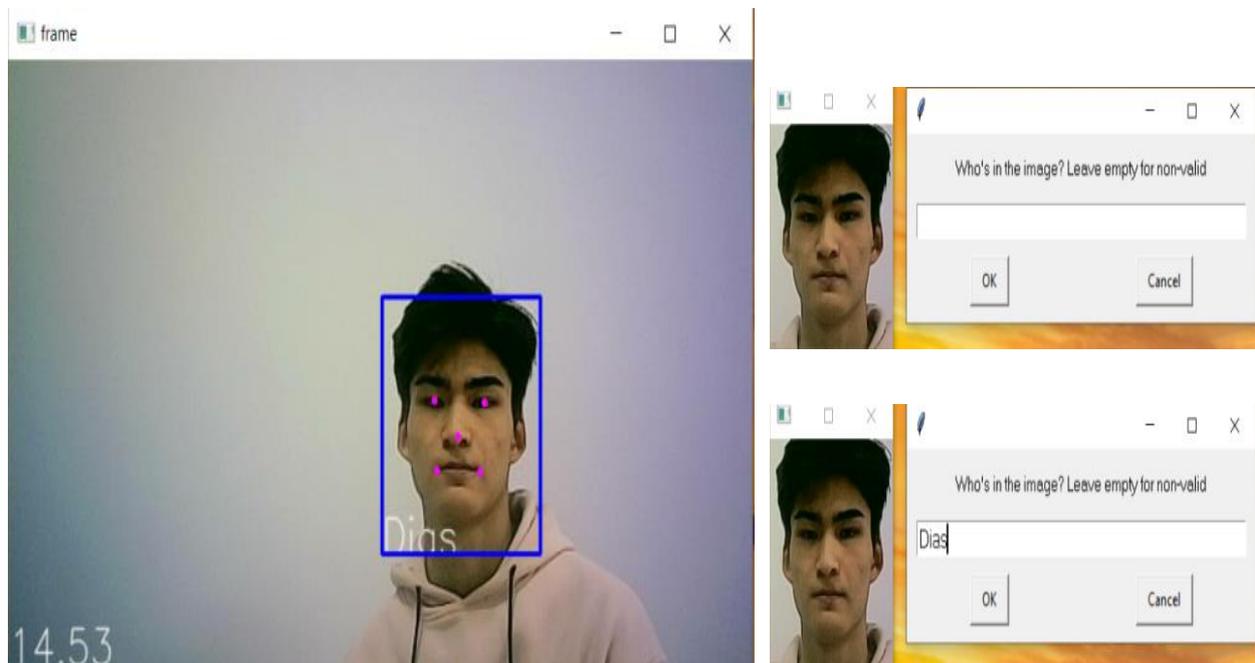


Figure 5. Creating a new object.

These lines (Algorithm 5) identify the object if it was previously saved and its id, i.e., the name of the object (Algorithm 5).

Algorithm 5 Object identification

```

1:     def detect_id_faces(self, image_paths):
2:         aligned_images = []
3:         id_image_paths = []
4:         for image_path in image_paths:
5:             image = cv2.imread(os.path.expanduser(image_path), cv2.IMREAD_COLOR)
6:             image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
7:             face_patches, _, _ = detect_and_align.detect_faces(image, self.mtcnn)
8:             if len(face_patches) > 1:
9:                 print("Warning: Found multiple faces in id image: %s" % image_path
10:                    + "\nMake sure to only have one face in the id images".
11:                    + "If that is the case then it's a false positive detection and"
12:                    + "you can solve it by increasing the thresholds of the cascade network")
13:             aligned_images = aligned_images + face_patches
14:             id_image_paths += [image_path] * len(face_patches)
15:             path = os.path.dirname(image_path)
16:             self.id_names += [os.path.basename(path)] * len(face_patches)
17:         return np.stack(aligned_images), id_image_paths

```

Results are shown in Figure 6.

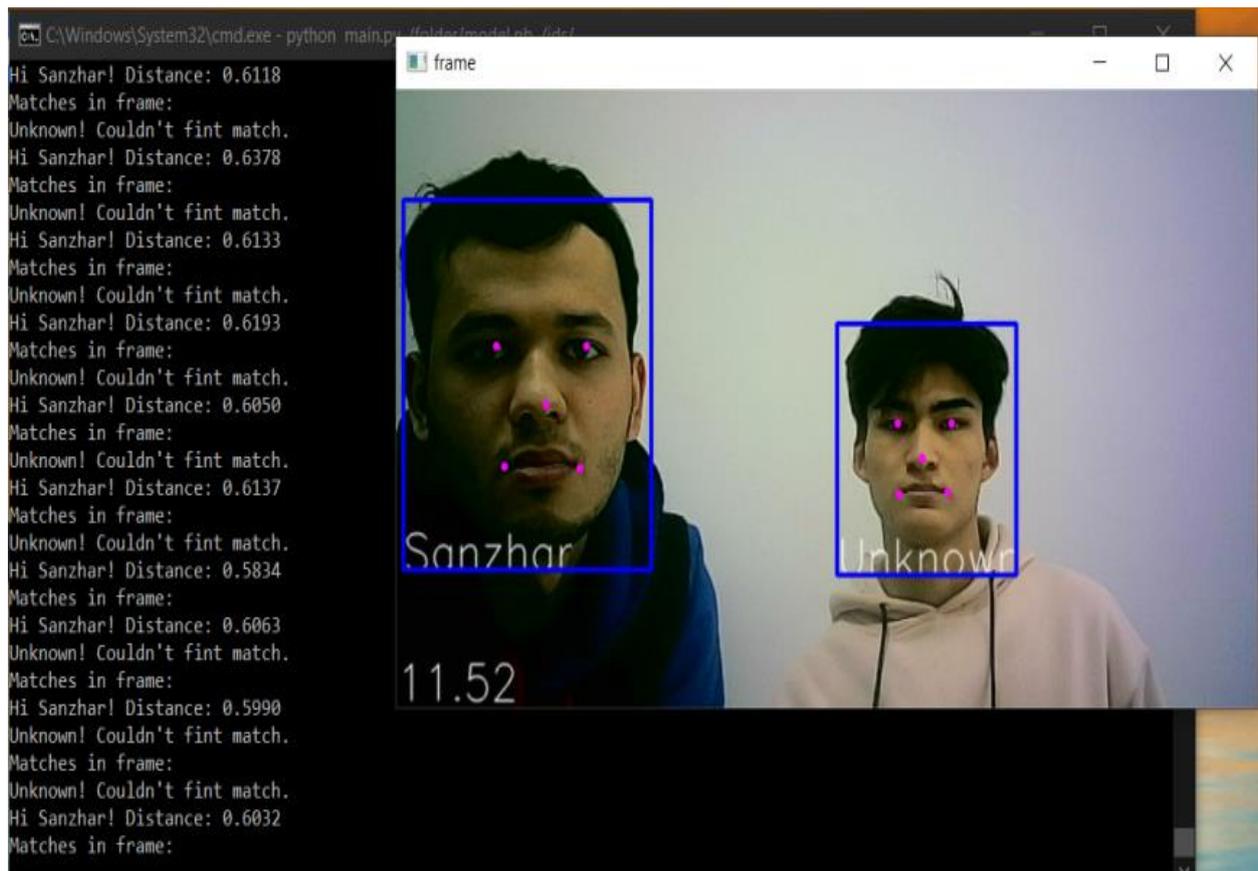


Figure 6. Object identification in the video stream.

We determined the presence of folders using `self`, `id_folder`, `mtcnn`, `sess`, `embeddings`, `images_placeholder`, `phase_train_placeholder`, `distance_treshold` algorithms. We determined the distance between the object and the camera, and also identified the object using Algorithm 5.

We asked some students to look up and down to take 170 pictures. We also looked at five different facial expressions of students during a photo shoot in a video stream, eight types of accessories (five glasses and three caps) and ten lighting directions. In Front image (FI), the students looked directly at the camera. The database included images of students looking up, down, horizontally, and the direction of head rotation in these images was within $[-75^\circ; +75^\circ]$ degrees (Figure 7). The detection results are shown in Table 3. Table 3 shows that the system successfully identified 7898 facial features; out of 7898 images (only one person in each picture) there were only 74 false ones. Analyzing certain cases of errors, we can conclude that the accuracy decreases primarily due to images with noise and poor lighting quality.



Figure 7. Results of face recognition in some images (a), (d) belonging; (b,c,e) different lighting conditions, (f): different angles.

Table 3. False positives and detection frequency from the dataset.

Data Set	Faces	False Alarms	Detection Rates
Front image (FI)	4000	39	98.95%
Image looking down (ILD)	1024	9	94.23%
Image looking up (ILU)	1254	11	96.12%
Image horizontally (IH)	1620	15	97.02%
Total	7898	74	96.85%

The highest hit percentage is in the Front image (FI) dataset, as all faces are frontal. In Image looking down (ILD), Image looking up (ILU) and Image horizontally (IH), there are facial expressions, accessories and different lighting options. It is these changes that lead to more false alarms and fewer hits in the Front image (FI) dataset. It can be concluded that finding images that look down is somewhat more difficult for the system than images that look up.

The main idea is to take into account the statistical relationships between the location of the anthropometric points of the face. According to their relative position, faces are compared. The human eye system is mobile. The eyeballs make different movements when looking at different objects. When looking up, the apples of the eyes are open; when looking down, the apples of the eyes are half-closed or more. This affects the accuracy of face recognition in a certain way.

The accuracy of face detection depends on the variability of the training set. Most face orienting and alignment models work well when viewed from the front. Front view detection can approach 98.6% (CNN, CNN+Haar) accuracy at -15° . Accuracy drops more than 50% as you approach the corners ($\alpha = -75^\circ$ and $\alpha = +75^\circ$). At angles above $+75^\circ$, -75° , the accuracy of face detection is not determined (Table 4). The source code

created by our team still needs to be improved. The CNN + Haar method generally works better than traditional statistical methods. Detection accuracy depends on viewing angles. Increasing focal length generally improves landmark and alignment performance at the cost of less projection distortion. Based on experimental results, all methods perform best from frontal viewing angles, as expected. It is also interesting to note that the rolloff slope for performance degradation caused by shorter focal lengths (wider field of view) is smaller for CNN-based methods and the CCN + Haar, SupportVector approach. This is probably due to the fact that the CNN, SupportVector methods are based on image features, while the Haar method emphasizes local image features more specifically.

Table 4. Face detection results at different angles.

Method/Degree	−75°	−45°	−30°	−15°	+15°	+30°	+45°	+75°	Average Value
CNN	N/A	80.0%	96.7%	98.6%	97.2%	97.5%	74.0%	N/A	90.6%
Support Vector	N/A	78.3%	92.2%	97.4%	97.5%	87.0%	83.1%	N/A	89.2%
Haar-based	N/A	76.2%	82.8%	88.5%	86.0%	82.6%	77.4%	N/A	82.2%
CNN+Haar (our model)	N/A	82.3%	97.6%	98.6%	98.0%	97.9%	76.7%	N/A	91.8%

Due to the speed of image processing, ease of implementation and the minimum cost of the technical requirements for the system, we chose the Viola–Jones method and the CNN method. Using the CNN method with the Viola–Jones method allows you to choose algorithms and methods that provide high accuracy and minimize false positives when solving face recognition problems. Haar cascades in Open CV computer vision were used for face detection (cascadePath = “haarcascade_frontalface_default.xml”, faceCascade = cv2.CascadeClassifier(cascadePath). Local binary templates were used for recognition (recognizer = cv2.createLBPHFaceRecognizer(1,8,8,8,123)).

We used the ROC curve to evaluate the CNN, Haar-based, Support Vector, CNN + Haar methods. ROC analysis is a graphical method for assessing the performance of a binary classifier and selecting a discrimination threshold for class separation. The ROC curve shows the relationship between FPR false alarm probability and TPR true positive probability. With increasing sensitivity, the reliability of recognition of positive observations increases, but at the same time, the probability of false alarms increases.

The accuracy of verification on faces is calculated. The ROC results for the CNN, Haar-based, Support Vector, CNN + Haar methods are presented in Table 5. Figure 8 shows the results of the ROC curves for four methods trained by the described methods. It can be seen from the graph that the classifier trained according to the proposed CNN + Haar method outperforms the classifier trained in the classical way for all FPR values. At the same time, the TPR value increased by an average of 0.0078 in absolute value, or by 1.2% in relative terms. The maximum increase in TPR was 0.01 in absolute value, or 1.9%. Note that since only a sample containing singular point samples (left eye center, right eye center, nose tip, left mouth corner, right mouth corner) is used to train this classifier, the unification of background points does not affect the result. Therefore, only the classifier trained by the classical method and the classifier trained by unifying singular points participated in the comparison. The CNN algorithm can process up to 50 frames per second on the CPU (in a single thread) and more than 350 frames per second on the GPU, and thus, is one of the fastest face detection algorithms at the moment.

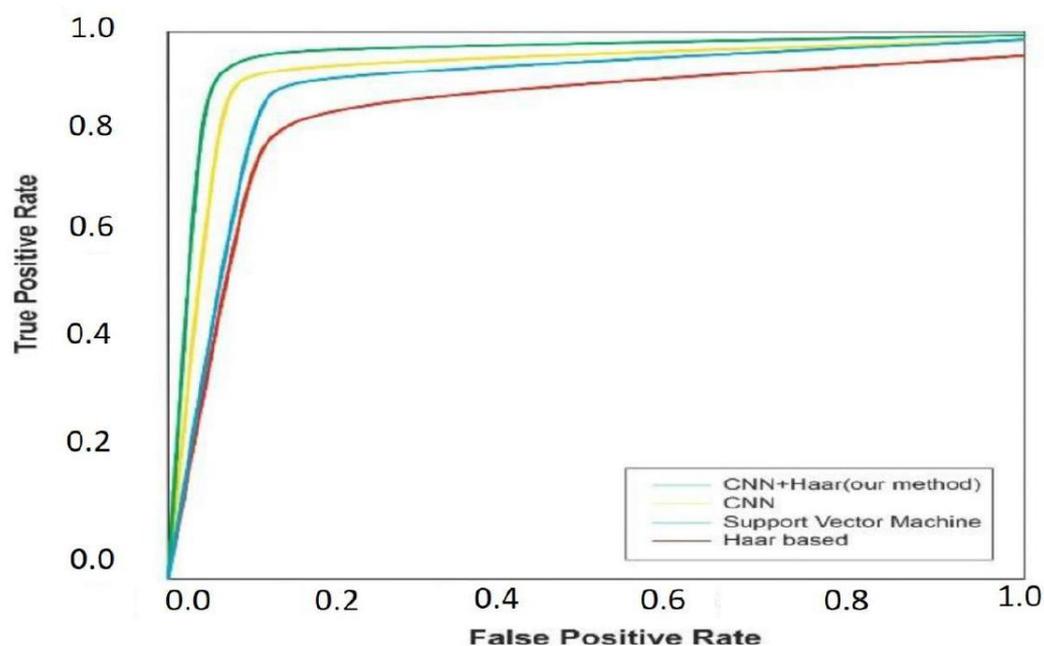


Figure 8. ROC curves for various methods of estimation and face search in images.

On average, the CNN + Haar and CNN algorithms provide an increase in TPR by 0.0190 and 0.0176, respectively, in absolute terms, or by 2.7 and 3.8% in relative terms. The maximum increase in TPR was 0.0418 and 0.0394, respectively, or 3.7 and 3.8%. Haar-based algorithms, on the contrary, showed a worse result by an average of 0.0086, or 1.1%, and a maximum of 0.0183, or 2.3%. When using new approaches to train the classifier, it is possible to achieve an improvement in the quality of segmentation by up to 4% according to the TPR metric, and a reduction in FPR classification errors by up to 10%. When comparing the ROC curves, it can be seen that the CNN + Haar classifier shows results that are noticeably superior to the results of the Support Vector classifier over the entire main operating range of FPR values. The share of correct answers of CNN algorithms is 0.9783, Support Vector is 0.9395, Haar-based is 0.7895 and CNN+Haar is 0.9895 (Table 5).

Table 5. Efficiency of CNN model identification, Support Vector, Haar-based models, CNN + Haar models.

Model	Accuracy	Sensitivity (Recall)	Specificity	Precision	F1-Score
CNN	0.9783	0.8772	0.9193	0.9776	0.9246
Support Vector	0.9395	0.7368	0.8421	0.8235	0.7777
Haar-based	0.7895	0.7362	0.8426	0.8237	0.8077
CNN + Haar (our model)	0.9895	0.8865	0.9429	0.9235	0.9046

The new method of training binary classifiers improves image quality by up to 4% by TPR (true positive bet result) and reduces segmentation errors by up to 10% by FPR (false positive) compared to the classical approach. The results of the proposed method and recommendations for its use can be formulated as follows:

1. When comparing binary classifiers, training and testing should be carried out on the same training and test sets.
2. The size of the test sample does not affect the test results.
3. When regenerating samples, the results practically do not change.
4. Test samples must be correctly composed so that the ROC curve does not contain duplicate points of the same class, as this leads to sharp jumps in TPR values.

4. Discussion

Recommendations are discussed for improving the algorithm to obtain a given accuracy. Since computer vision is a major research area, many researchers have invested a lot of energy in it, and hundreds of papers related to it are published every year. After analyzing the results of testing the algorithm, we can say that some of the errors occur due to noise in the image, which, when scaled, gives the effect of “blurring”. A more common reason is the incomplete invariance of the algorithm to the level of illumination. From the consideration of the value of the outputs of the first layer of neurons, it becomes clear that after training, a significant part of the neurons begins to respond only to illumination, clearly separating the background from the object.

The next group of errors occurs for images containing a face with a significant rotation or inclination (provided that there was no such image in the training sample for this class). To reduce the influence of noise and improve the overall accuracy at the preprocessing stage, the wavelet transform is proposed to use. To increase the stability of the algorithm to the quality of lighting, the normalization method is usually used. Its essence is to bring the statistical characteristics of the image (mathematical expectation and dispersion of pixel values) to fixed values.

Some problems of automated face recognition have not yet been fully resolved. In recent years, a number of different approaches to processing, localization and recognition of objects have been proposed, such as the principal component method, neural networks, evolutionary algorithms, the AdaBoost algorithm, the support vector machine, etc. However, these object recognition approaches lack accuracy, reliability and speed in a complex real-world environment characterized by the presence of noise in images and video sequences.

Difficulties arising in face recognition include overlapping faces, various head turns and tilts, variability of spectra, illumination intensities and angles, and facial expressions. Some of the errors occur due to the noise in the image, which, when scaled, gives the effect of “blurring”. A more common reason is the incomplete invariance of the algorithm to the level of illumination. A certain group of errors occurs for images containing a face with a significant rotation or inclination (provided that there was no such image in the training sample for this class). A certain group of errors occurs for images containing a face with a significant rotation or inclination (provided that there was no such image in the training sample for this class). The methods used to solve the problem of face and gesture recognition should provide acceptable recognition accuracy and high processing speed of video sequences. Thus, it is necessary to improve methods and algorithms for recognizing faces and gestures on static images and moving objects on video sequences in real time.

5. Conclusions

In the work provided, the main work of a neural network in Python using different face recognition algorithms is considered, the necessary approaches are implemented and an algorithm is selected. The algorithm can be widely used in the proctoring system and other automatic recognition systems. The article substantiates the rationality of using face detection technologies in order to increase the reliability of proctoring systems. The recognition system makes it possible to adjust the corresponding functions: managing the assignment of information from the camera, performing face detection in the frame, real-time shooting, where the object is located and real-time user recognition. In real life, facial recognition is widely used and has extremely wide prospects for development. While face recognition and face detection technologies are constantly evolving, the accuracy and amount of computation is constantly improving. Furthermore, the application or algorithms change frequently. The registration-based software for accessing a video stream analyzes its frames and offers the effects needed to detect a person in the video stream.

In this article, we reviewed mathematical models and face recognition algorithms. We compared existing methods such as Principal Component Analysis, Independent Component Analysis, Active Shape Model, Hidden Markov Model, Support Vector Machines and Neural Network Method. We identified the main disadvantages and advantages of

these methods. We wrote a face recognition algorithm based on the CNN method and Viola–Jones and faced some problems related to photographs taken from different angles. Having examined the photographs taken at different angles, we found out that the accuracy of surface detection at angles above +750, −750 is not defined. It was found that the accuracy of face detection depends on the viewing angles.

The ROC curve was used to evaluate the CNN, Haar-based, Support Vector, CNN + Haar methods. When comparing the ROC curves, it can be seen that the CNN + Haar classifier shows results that are noticeably superior to the results of other classifiers over the entire main range of FPR values. The percentage of correct answers of the algorithms showed the following results: CNN—0.9783, Support Vector—0.9395, Haar-based—0.7895, CNN + Haar—0.9895. Human movement algorithms and eye movement algorithms can be complementary to detect some abnormalities in the testing process. In subsequent studies, we will consider algorithms for capturing human movements and eye movements or eye tracking models in video streams.

The results of the test program showed that the targeted use of the original algorithm makes it possible to effectively recognize surfaces in digital images and video streams. Human movement algorithms and eye movement algorithms can be complementary to detect some abnormalities in the testing process. In subsequent studies, we will consider algorithms for capturing human movements and eye movements or eye tracking models in video streams.

Author Contributions: Conceptualization, A.N., A.S. (Anargul Shaushenova), S.N., Z.Z., M.O., Z.M., A.S. (Alexander Semenov) and L.M.; methodology, A.N., A.S. (Anargul Shaushenova), S.N., Z.Z., M.O., Z.M., A.S. (Alexander Semenov) and L.M.; formal analysis, A.N., A.S. (Anargul Shaushenova), S.N., Z.Z., M.O. and Z.M.; investigation, A.N., A.S. (Anargul Shaushenova), S.N., M.O. and Z.M.; resources, A.N., A.S. (Anargul Shaushenova), S.N., Z.Z., M.O. and Z.M.; writing—original draft preparation, A.N., A.S. (Anargul Shaushenova), S.N., M.O. and Z.M.; writing—review and editing, A.N., A.S. (Anargul Shaushenova), Z.Z., M.O. and Z.M.; visualization, A.N., Z.Z., M.O. and Z.M.; supervision, A.N., A.S. (Anargul Shaushenova), S.N., M.O. and Z.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by project No. AP09259657 Research and development of an automated proctoring system for monitoring students’ knowledge in the context of distance learning.

Institutional Review Board Statement: Ethical review and approval have been cancelled for this study, as no experiments were performed in this research work that could harm humans, animals, or wildlife. Although the students were involved in the study of the pattern recognition system, no harm was done to their lives, health, or individual rights.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not applicable.

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

References

1. Nurpeisova, A.; Mauina, G.; Niyazbekova, S.; Jumagaliyeva, A.; Zholtmukhanova, A.; Tyurina, Y.G.; Murtuzaliev, S.; Maisigova, L.A. Impact of R and D expenditures on the country’s innovative potential: A case study. *Entrep. Sustain. Issues* **2020**, *8*, 682–697. [[CrossRef](#)]
2. Niyazbekova, S.; Yessymkhanova, Z.; Kerimkhulle, S.; Brovkina, N.; Annenskaya, N.; Semenov, A.; Burkaltseva, D.; Nurpeisova, A.; Maisigova, L.; Varzin, V. Assessment of Regional and Sectoral Parameters of Energy Supply in the Context of Effective Implementation of Kazakhstan’s Energy Policy. *Energies* **2022**, *15*, 1777. [[CrossRef](#)]
3. Maisigova, L.A.; Niyazbekova, S.U.; Isayeva, B.K.; Dzholdosheva, T.Y. Features of Relations between Government Authorities, Business, and Civil Society in the Digital Economy. *Stud. Syst. Decis. Control* **2021**, *314*, 1385–1391.
4. Evmenchik, O.S.; Niyazbekova, S.U.; Seidakhmetova, F.S.; Mezentceva, T.M. The Role of Gross Profit and Margin Contribution in Decision Making. *Stud. Syst. Decis. Control* **2021**, *314*, 1393–1404.
5. Niyazbekova, S.; Grekov, I.; Blokhina, T. The influence of macroeconomic factors to the dynamics of stock exchange in the Republic of Kazakhstan. *Econ. Reg.* **2016**, *12*, 1263–1273. [[CrossRef](#)]

6. Niazbekova, S.; Aetdinova, R.; Yerzhanova, S.; Suleimenova, B.; Maslova, I. Tools of the Government policy in the Area of Controlling poverty for the purpose of sustainable development. In Proceedings of the 2020 2nd International Conference on Pedagogy, Communication and Sociology (ICPCS 2020), Bangkok, Thailand, 6–7 January 2020. [[CrossRef](#)]
7. Shaushenova, A.; Zulpykhar, Z.; Zhumasseitova, S.; Ongarbayeva, M.; Akhmetzhanova, S.; Mutalova, Z.; Niyazbekova, S.; Zueva, A. The Influence of the Proctoring System on the Results of Online Tests in the Conditions of Distance Learning. *Ad Alta J. Interdiscip. Res.* **2021**, *11*, 250–256.
8. Niyazbekova, S.; Moldashbayeva, L.; Kerimkhulle, S.; Jazykbayeva, B.; Belousova, E.; Suleimenova, B. Analysis of the development of renewable energy and state policy in improving energy efficiency. *E3S Web Conf.* **2021**, *258*, 11011. [[CrossRef](#)]
9. Karim, M.N.; Kaminsky, S.E.; Behrend, T.S. Cheating, reactions, and performance in remotely proctored testing: An exploratory experimental study. *J. Bus. Psychol.* **2014**, *29*, 555–572. [[CrossRef](#)]
10. Brothen, T.; Peterson, G. Online exam cheating: A natural experiment. *Int. J. Instr. Technol. Distance Learn.* **2012**, *9*, 15–20.
11. Dunn, T.P.; Meine, M.F.; McCarley, J. The remoteproctor: An innovative technological solution for onlinecourse integrity. *Int. J. Technol. Knowl. Soc.* **2010**, *6*, 1–7. [[CrossRef](#)]
12. Lilley, M.; Meere, J.; Barker, T. Remote live invigilation: A pilot study. *J. Interact. Media Educ.* **2016**, *1*, 1–5. [[CrossRef](#)]
13. Tomasi, L.F.; Figiel, V.L.; Widener, M. I have got my virtual eye on you: Remote proctors and academic integrity. *Contemp. Issues Educ. Res.* **2009**, *2*, 31–35. [[CrossRef](#)]
14. Wright, N.A.; Meade, A.W.; Gutierrez, S.L. Using invariance to examine cheating in unproctored ability tests. *Int. J. Sel. Assess.* **2014**, *22*, 12–22. [[CrossRef](#)]
15. Viola, P.; Jones, M.J. Robust real-time face detection. *Int. J. Comput. Vis.* **2004**, *57*, 137–154. [[CrossRef](#)]
16. Viola, P.; Jones, M.J. 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), Kauai, HI, USA, 8–14 December 2001; pp. I-511–I-518.
17. Mestetsky, L.M. *Mathematical Methods of Pattern Recognition*; Moscow State University: Moscow, Russia, 2004; p. 85.
18. Amirgaliyev, Y.; Shamiluulu, S.; Serek, A. Analysis of Chronic Kidney Disease Dataset by Applying Machine Learning Methods. In Proceedings of the 2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT), Almaty, Kazakhstan, 17–19 October 2018; pp. 1–4. [[CrossRef](#)]
19. Khoroshilov, A.D.A.; Musabaev, R.R.; Kozlovskaya, Y.D.; Nikitin, Y.A.; Khoroshilov, A.A. Automatic Detection and Classification of Information Events in Media Texts. *Autom. Doc. Math. Linguist.* **2020**, *54*, 202–214. [[CrossRef](#)]
20. Dyusembayev, A.; Grishko, M. On correctness conditions for the algebra of recognition algorithms with the initial set of μ -operators over the set of problems with binary information. *Rep. Acad. Sci.* **2018**, *482*, 128–131.
21. Tippins, N.T.; Beaty, J.; Drasgow, F.; Gibson, W.M.; Pearlman, K.; Segall, D.O.; Shepherd, W. Unproctored internet testing in employment settings. *Pers. Psychol.* **2006**, *59*, 189–225. [[CrossRef](#)]
22. Boonsuk, W.; Saisin, Y. The Application of a Face Recognition System for the Criminal Database. *J. Appl. Inform. Technol.* **2021**, *3*, 14–21. [[CrossRef](#)]
23. Turk, M.; Pentland, A. Eigenfaces for Recognition. *J. Cogn. Neurosci.* **1991**, *3*, 71–86. [[CrossRef](#)]
24. Grother, P.; Phillips, P.J. Models of large population recognition performance. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), Washington, DC, USA, 27 June–2 July 2004; Volume 2, pp. II-68–II-75.
25. Xiaoli, Y.; Guangda, S.; Jiansheng, C.; Nan, S.; Xiaolong, R. Large Scale Identity Deduplication Using Face Recognition Based on Facial Feature Points. In Proceedings of the 6th Chinese Conference on Biometric Recognition: CCBR 2011, Beijing, China, 3–4 December 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 25–32.
26. Hasan, M.K.; Ahsan, M.S.; Newaz, S.H.S.; Lee, G.M. Human Face Detection Techniques: A Comprehensive Review and Future Research Directions. *Electronics* **2021**, *10*, 2354. [[CrossRef](#)]
27. Rahman, F.H.; Newaz, S.S.; Au, T.W.; Suhaili, W.S.; Mahmud, M.P.; Lee, G.M. EnTruVe: ENergy and TRUst-aware Virtual Machine allocation in VEHICLE fog computing for catering applications in 5G. *Future Gener. Comput. Syst.* **2021**, *126*, 196–210. [[CrossRef](#)]
28. Wang, M.; Deng, W. Deep face recognition: A survey. *Neurocomputing* **2021**, *429*, 215–244. [[CrossRef](#)]
29. Huang, D.; Shan, C.; Ardabilian, M.; Wang, Y.; Chen, L. Local binary patterns and its application to facial image analysis: A survey. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2011**, *41*, 765–781. [[CrossRef](#)]
30. Sanchez-Moreno, A.S.; Olivares-Mercado, J.; Hernandez Suarez, A.; Toscano-Medina, K.; Sanchez-Perez, G.; Benitez-Garcia, G. Efficient Face Recognition System for Operating in Unconstrained Environments. *J. Imaging* **2021**, *7*, 161. [[CrossRef](#)]
31. Zhu, Q.; He, Z.; Zhang, T.; Cui, W. Improving Classification Performance of Softmax Loss Function Based on Scalable Batch Normalization. *Appl. Sci.* **2020**, *10*, 2950. [[CrossRef](#)]
32. Sparks, D.L. The brainstem control of saccadic eye movements. *Nat. Rev. Neurosci.* **2002**, *3*, 952–964. [[CrossRef](#)]
33. Singh, S.; Ahuja, U.; Kumar, M.; Kumar, K.; Sachdeva, M. Face mask detection using yolov3 and faster r-cnn models: COVID-19 environment. *Multimed. Tools Appl.* **2021**, *80*, 19753–19768. [[CrossRef](#)]
34. Hong, J.H.; Kim, H.; Kim, M.; Nam, G.P.; Cho, J.; Ko, H.S.; Kim, I.J. A 3D model-based approach for fitting masks to faces in the wild. In Proceedings of the 2021 IEEE International Conference on Image Processing (ICIP), Anchorage, AK, USA, 19–22 September 2021; pp. 235–239.

35. Nefian, A.V.; Hayes, M.H. Maximum likelihood training of the embedded HMM for face detection and recognition. In Proceedings of the 2000 International Conference on Image Processing (Cat. No. 00CH37101), Vancouver, BC, Canada, 10–13 September 2000; Volume 1, pp. 33–36.
36. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4690–4699.
37. Li, Y.; Guo, K.; Lu, Y.; Liu, L. Cropping and attention based approach for masked face recognition. *Appl. Telligence* **2021**, *51*, 3012–3025. [[CrossRef](#)]
38. Priya, G.N.; Banu, R.W. Occlusion invariant face recognition using mean based weight matrix and support vector machine. *Sadhana* **2014**, *39*, 303–315. [[CrossRef](#)]
39. Sikha, O.K.; Bharath, B.B. VGG16-random fourier hybrid model for masked face recognition. *Soft Comput.* **2022**, *in press*. [[CrossRef](#)]
40. Ding, C.; Tao, D. Trunk-branch ensemble convolutional neural networks for video-based face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 1002–1014. [[CrossRef](#)] [[PubMed](#)]
41. Zhang, X.; Gao, Y. Face recognition across pose: A review. *Pattern Recognit.* **2009**, *42*, 2876–2896. [[CrossRef](#)]
42. Zou, X.; Kittler, J.; Messer, K. Illumination invariant face recognition: A survey. In Proceedings of the IEEE International Conference on Biometrics: Theory, Applications, and Systems, Crystal City, VA, USA, 27–29 September 2007.
43. Tistarelli, M.; Bicego, M.; Grosso, E. Dynamic face recognition: From human to machine vision. *Image Vis. Comput.* **2009**, *27*, 222–232. [[CrossRef](#)]
44. Wang, H.; Wang, Y.; Cao, Y. Video-based face recognition: A survey. *Int. J. Comput. Inf. Eng.* **2009**, *3*, 293–302.
45. Matta, F.; Dugelay, J. Person recognition using facial video information: A state of the art. *J. Vis. Lang. Comput.* **2009**, *20*, 180–187. [[CrossRef](#)]
46. Hadid, A.; Dugelay, J.; Pietikainen, M. On the use of dynamic features in face biometrics: Recent advances and challenges. *Signal Image Video Process.* **2011**, *5*, 495–506. [[CrossRef](#)]
47. Barr, J.R.; Bowyer, K.W.; Flynn, P.J.; Biswas, S. Face recognition from video: A review. *Int. J. Pattern Recognit. Artif. Intell.* **2012**, *26*. [[CrossRef](#)]
48. Scheenstra, A.; Ruifrok, A.; Velkamp, R.C. A survey of 3D face recognition methods. In Proceedings of the International Conference on Audio and Video Based Biometric Person Authentication, Hilton Rye Town, NY, USA, 20–22 July 2005; pp. 891–899.
49. Bowyer, K.W.; Chang, K.; Flynn, P. A survey of approaches and challenges in 3d and multi-modal 3d + 2d face recognition. *Comput. Vis. Image Underst.* **2006**, *101*, 1–15. [[CrossRef](#)]
50. Zhou, H.; Mian, A.; Wei, L.; Creighton, D.; Hossny, M.; Nahavandi, S. Recent advances on singlemodal and multimodal face recognition: A survey. *IEEE Trans. Human Mach. Syst.* **2014**, *44*, 701–716. [[CrossRef](#)]
51. Scherhag, U.; Rathgeb, C.; Merkle, J.; Breithaupt, R.; Busch, C. Face recognition systems under morphing attacks: A survey. *IEEE Access* **2019**, *7*, 23012–23026. [[CrossRef](#)]
52. Wiedenbeck, S.; Waters, J.; Birget, J.C.; Brodskiy, A.; Memon, N. PassPoints: Design and longitudinal evaluation of a graphical password system. *Int. J. Hum. Comput. Stud.* **2005**, *63*, 102–127. [[CrossRef](#)]
53. Schneier, B. Two-factor authentication: Too little, too late. *ACM Commun.* **2005**, *48*, 136. [[CrossRef](#)]
54. Kasprowski, P. Human Identification Using Eye Movements, Faculty of Automatic Control, Electronics and Computer Science. Ph.D. Thesis, Silesian University of Technology, Gliwice, Poland, 2004; p. 111.
55. Josephson, S.; Holmes, M.E. Visual attention to repeated internet images: Testing the scanpath theory on the world wide web. In Proceedings of the 2002 Symposium on Eye Tracking Research and Applications (ETRA 02), New Orleans, LA, USA, 25–27 March 2002; pp. 43–49.
56. Williams, J.M. Biometrics or biohazards? In Proceedings of the 2002 Workshop on New Security Paradigms, Virginia Beach, VA, USA, 23–26 September 2002; pp. 97–107.
57. Komogortsev, O.; Khan, J. Eye Movement Prediction by Kalman Filter with Integrated Linear Horizontal Oculomotor Plant Mechanical Model. In Proceedings of the 2008 Symposium on Eye Tracking Research and Applications Symposium, Savannah, GA, USA, 26–28 March 2008; pp. 229–236.
58. Salvucci, D.D.; Goldberg, J.H. Identifying fixations and saccades in eye tracking protocols. In Proceedings of the Eye Tracking Research and Applications Symposium, Palm Beach Gardens, FL, USA, 6–8 November 2000; pp. 71–78.
59. Komogortsev, O.; Khan, J. Eye movement prediction by oculomotor plant Kalman filter with brainstem control. *J. Control Theory Appl.* **2009**, *7*, 14–22. [[CrossRef](#)]
60. Koh, D.H.; Gowda, S.A.M.; Komogortsev, O.V. Input evaluation of an eye-gaze-guided interface: Kalman filter vs. velocity threshold eye movement identification. In Proceedings of the 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems, Pittsburgh, PA, USA, 15–17 July 2009; pp. 197–202.