UDC 004.932 **doi: 10.20998/2522-9052.2018.3.02**

K. Dergachov, L. Krasnov, O. Cheliadin, A. Zymovin

National Aerospace University – Kharkiv Aviation Institute, Kharkiv, Ukraine

ADAPTIVE ALGORITHMS OF FACE DETECTION AND EFFECTIVENESS ASSESSMENT OF THEIR USE

Subject of research is the detection and recognition of faces. **The purpose** of this work is creation of modified algorithms of face detection, which are providing automatic brightness stabilization of the analyzed image regardless of brightness level. A technique is proposed for assessing the effectiveness of their work in comparison with the classical algorithm. **Research methods**. We will dwell in more detail on the first part of the problem − face detection and recognition. In the meantime, the most popular method used for searching the face area on an image is the Viola-Jones method, which is popular because of its known high speed and efficiency. It is based on an integral image representation, on the method of constructing classifiers based on adaptive boosting algorithm (AdaBoost) and on the combination classifiers in cascade structure method. The Viola-Jones method is firstly using cascades of wavelets (primitives) - Haar features. All of the above made it possible to build a face detector that works in real-time with a fairly high quality. However, there are a lot of disturbing factors, which are limiting the efficiency of such algorithm work. The major of them are spacial face position ambiguity on the analised image and poor quality of stage lighting. **The results of the study.** The adaptive algorithms of face detection and recognition on digital images and video sequences in real-time, based on the Viola-Jones method, are suggested. An automatic stabilization of frame brightness is additionally added to the classical structure of such algorithms to compensate an effect of changes in the stage illumination level on quality of face detection. The structure of the algorithms is described and the software developed in Python programming language for a face detection and recognition using OpenCV library resources. Video data is processed in real time. An original method for the efficient estimating of the algorithm based on the criterion of the maximum probability of faces and their main elements (eye, nose, mouth) correct detection is proposed and implemented programmatically. The results of work of classic and suggested algorithms are compared. The examples of work and testing of software are given. **Conclusion.** The use of the obtained results allows to improve the quality of work and the reliability of the results when recognizing faces in different systems.

Keywords: face detection and recognition; the Viola-Jones method; an automatic stabilization of frame brightness; the probability of face and its main elements correct detection.

Introduction

Practical needs in face recognition are constatly increasing in different fields. The examples of the above are: security and face-control in the segment of mass social events and entertainment, search of potentially dangerous visitors suspected in terrorist intenstions, bank cards and electronic payment verification and a lot of other actual and useful tasks. All of these is permanently attracts an attention to the fundament task of computer vision − veracious face detection and recognition on digital images and video sequences in real-time. It is necessary to steadily increase the quality of video data processing and improve face detection and recognition algorithms in complicated conditions.

Face detection process is generally divided in two parts: a face search and recognition on an image and comparing found face with faces stored in databases. In our work, we will dwell in more detail on the first part of the problem − face detection and recognition. In the meantime the most popular method used for searching the face area on an image is the Viola-Jones method [1, 2], which is popular because of its known high speed and efficiency. The essence of the method is widely known. It is based on an integral image representation, on the method of constructing classifiers based on adaptive boosting algorithm (AdaBoost) and on the combination classifiers in cascade structure method. The Viola-Jones method is firstly using cascades of wavelets (primitives) - Haar features, representing a breakdown of a given rectangular area into sets different types of rectangular subregions. All of the above made it possible to build a face detector that works in real-time with a fairly high quality. However, there are a lot of disturbing factors, which are limiting the efficiency of such algorithm work. The major of them are spacial face position ambiguity on the analyzed image and poor quality of stage lighting.

The purpose of this work is creation of modified algorithms of face detection, which are providing automatic brightness stabilization of the analyzed image regardless of brightness level. A technique is proposed for assessing the effectiveness of their work in comparison with the classical algorithm.

Methods and tools for constructing new algorithms

Let's consider the methods for solving the task and the means involved.

Using tools. When creating algorithms, the authors used the Python programming language and OpenCV [3, 4] library resources. This choice is due to open access to the software products and their compatibility with operation systems, such as Windows, Linux and Android. In this approach, the work of algorithms is easily implemented on a single-board computer Raspberry Pi. The peculiarity of writing codes in Python is the formation of the necessary programming capabilities by connecting external libraries. To solve our task, the imported resources of the OpenCV [6] library are relatively small:

```
import cv2,
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches.
```

The number of programming resources being used does not have a significant load on the processor and creates good prerequisites for processing data in real-time. For a significant increase in the speed with the processing of video data, all the basic procedures are implemented using standard OpenCV library functions, optimized by the library developers even at the stage of their creation.

Stabilizing the contrast of video data. Detection and registration of individuals faces using various video sensors usually occurs against a background of various types of interference with a constant dynamic change in the background of the observed scene. One of the dominant negative factors is variability of illumination. Note that this can be both rapidly changing lighting conditions, as well as its slow changes due to, for example, the onset of twilight. All this leads to poorly controlled variations in the contrast of the frames, and, consequently, to a deterioration in the quality of processing. Recall that the brightness values for video data is usually changing in the range of numbers $0 \div 255$. This corresponds to the uint8 data format. To overcome these difficulties (poorly controlled changes in the contrast of the frames depending on the illumination), the authors suggested the original video sequence from the RGB color space to be converted into the YUV space using the function

```
img_yuv = cv2.cvtColor
(img, cv2.COLOR_BGR2YUV).
```

The range of RGB values is [0 ÷ 255] for each component, and for the YUV color space, the ranges

- Y → [0 ÷ 255];
- U → [-112 ÷ 112];
- V → [-157 ÷ 157].

A distinctive feature of the YUV color space [5] is that it uses an explicit separation of information on brightness and color. Color is represented as three components - brightness (Y) and two color difference (U and V).

After the RGB frame of the video sequence has been transferred to the YUV color space, an equalization procedure (enhancement of the contrast) of only the Y component with the function

```
img_yuv[:,:,0] =
cv2.equalizeHist(img_yuv[:,:,0]),
```

and then the frame is converted back from the YUV format to the RGB format

```
img_output=cv2.cvtColor
(img_yuv,cv2.COLOR_YUV2BGR).
```

In this case, the color balance remains unchanged, since the color difference components U and V were not transformed.

Note that the transition from the RGB color space to the YUV space makes it easy to estimate the average level of brightness of the frame by component Y. The most objective and stable indicator in our opinion is MB (Medium Brightness), which is calculated as

$$MB = \frac{1}{M \cdot N} \cdot \sum_{i=0}^{N} \sum_{j=0}^{M} Y(i, j)$$

where *Y(i,j)* is a two-dimensional array of numbers that determine the brightness of pixels of a $M \times N$ frame image. This indicator is all the more useful because at low levels of frame brightness in video surveillance systems for face recognition it is convenient to use the procedure of its comparison with a preset threshold for automatic switching on / off of the scene illumination system. However, the use of the indicator of average brightness MB causes significant difficulties. In the OpenCV library, there is no optimized function for calculating the normalized average brightness of images, and cyclic summation of pixel brightness indicators throughout the frame (especially large ones) too slows down the algorithms and does not allow processing in real-time. Therefore, in the proposed algorithm, this indicator is not used, and control of the level of illumination of the stage is carried out adaptively on the basis of other indicators, the work of which will be described below.

Thus, the use of the equalization procedure ensures alignment of the brightness distribution histogram and brings the average brightness indicator of the image to the value (MB = 127) regardless of how this indicator was for the original image. This makes it possible to stabilize the level of average brightness of the frames, and consequently, to simplify the adjustment of the devices` parameters for detecting faces and their main elements (eyes, nose, mouth). Such processing makes the algorithm more resistant to external factors. An example of usage the equalization procedure in the process of face and eyes detection is shown in Fig. 1.

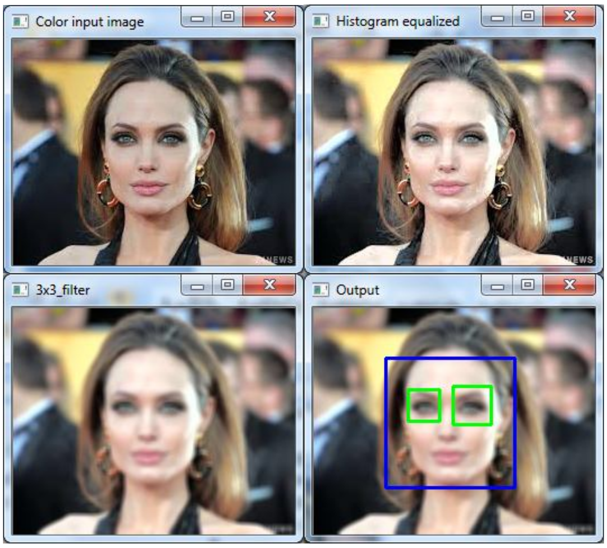


Fig. 1. Input image → histogram equalized → image after filtration → output image after recognition

Face detection and recognition based on the Viola-Jones method. As noted earlier, most modern recognition systems for searching and selecting faces are oriented to the use of the Viola-Jones method, which is based on the integrated representation of the image, the construction of classifiers based on the adaptive boosting algorithm, and the creation of classifiers in the form of a cascade structure. The most time-consuming process is learning the Haar cascades using the

AdaBoost machine learning algorithm. However, at the present time for Haar cascading classifiers there is a large number of already trained cascades, including the standard supply of the OpenCV library. Its installation package contains a whole set of ready-trained classifiers stored as files with an extension ".xml". In this set there are classifiers, both for face searching, and for its separate parts (eyes, mouth, nose) searching.

When synthesizing recognition algorithms, the use of standard (pre-trained) classifiers is most productive. In this regard, we consider it appropriate to provide a list of the main pre-trained classifiers for reference:

- haarcascade_eye_tree_eyeglasses.xml;
- haarcascade_fullbody.xml;
- haarcascade_mcs_lefteye.xml;
- haarcascade_profileface.xml;
- haarcascade_eye.xml;
- haarcascade_lefteye_2splits.xml;
- haarcascade_mcs_mouth.xml;
- haarcascade_righteye_2splits.xml;
- haarcascade_frontalface_alt2.xml;
- haarcascade_lowerbody.xml;
- haarcascade_mcs_nose.xml;
- haarcascade_smile.xml;
- haarcascade_frontalface_alt_tree.xml;
- haarcascade_mcs_eyepair_big.xml;
- haarcascade_mcs_rightear.xml;
- haarcascade_upperbody.xml;
- haarcascade_frontalface_alt.xml;
- haarcascade_mcs_eyepair_small.xml;
- haarcascade_mcs_righteye.xml;
- haarcascade_frontalface_default.xml;
- haarcascade_mcs_leftear.xml;
- haarcascade_mcs_upperbody.xml.

Without dwelling in detail on the purpose and properties of individual classifiers, we note that special interest among them are those that allow us to determine the position of the eyes. This is a very useful procedure, since it enables evaluation of the distance between the pupils. This further analysis allows to eliminate factors associated with the inclination of the head. Classifiers trained in the search for the eyes are very sensitive to the presence of glasses. In most cases, they fail, especially for glasses that completely hide the eyes. In such cases, a more effective eye detection classifier should be used «haarcascade_eye_tree_eyeglasses.xml», trained to look at the image of the eyes with glasses. It can be used as a backup option in cases of failure in the normal classifier.

The most effective methods are searching for the main elements in the already selected area of the face, as this locates the search region and shortens the analysis time, and also significantly reduces the probability of false positives. Selecting areas of interest for eye, nose and mouth detection requires optimizing the position and size of the area of interest for each element. An example of such a partitioning of the selected area into search zones of individual face elements in our algorithm is shown in Fig. 2.

Dimensions of the search areas of our algorithm were determined experimentally, but if necessary, they can be further optimized. Note that there are two ways of detecting the eyes - highlighting the common area of interest for two eyes (or the search area) and searching and detection of left and right eyes separately. In OpenCV, the appropriate classifiers are provided for this. Which of these methods is preferable, in our work will be determined experimentally.

The quality parameters of algorithms work and methods of their use. An objective criterion for the effectiveness of any algorithm for detecting faces and their elements in frames of a video sequence is the probability of correct face detection, provided it is present in the frame.

The procedure for detecting a face in the current frame using the appropriate cascading classifier, if successful, is completed by building a rectangle using the OpenCV function

```
cv2.rectangle(frame,(x,y)(x+w,y+h),
255,0,0),2).
```

Successful completion of face detection in the program code of the algorithm is necessary accompany the appearance of the event with a logical 1; otherwise, the appearance of a logical 0. In our work, in order to create a stable criterion for the quality of correct faces detection, the probability P_n, which is the result of the counting the number of successful detections in a sliding window with the size of 100 frames and normalized to the size of the window. At a standard video frame rate of 30/s, the window has a time constant of change P_n approximately equal to 3.3 s. The quality of the detector can be considered satisfactory for $P_n \geq 0{,}9$. According to the probability value of correct faces detection in the frame, it is easy to analyze the influence of various factors (including changes in illumination, geometric factors, etc.) on the quality of the faces detection and detection algorithms.

Similarly, the probability of correct detection of the eyes, nose, and mouth in the corresponding areas is calculated. These indicators are calculated as the conditional probability of such an event, provided that the entire face is correctly detected. With confidence that is acceptable for practical use, it is possible to consider events of individual elements of a face detection as an independent, and consequently, the probability of correctly detecting all elements is estimated as a product of these elements detection probabilities.

To adapt the algorithms for detecting faces proposed in our work to the conditions for changing the external illumination of the scene, it is suggested to organize two channels for processing video data (see the diagram in Fig. 3). In one of them, the original video stream is used directly for the detection of faces, and in the second channel, it is pre-equalized to bring the brightness of the frame to an average value. On the outputs of both processing channels in sliding windows with the size of 100 frames, the probabilities of the correct detection of faces and their elements are calculated, and then they are compared in the comparison block. Depending on the sign of the difference in probability ΔP_n with the help of the feedback loop, the results of detecting faces from the channel in which the lighting conditions are more comfortable for the operation of the face detection algorithm are transmitted to the system output.

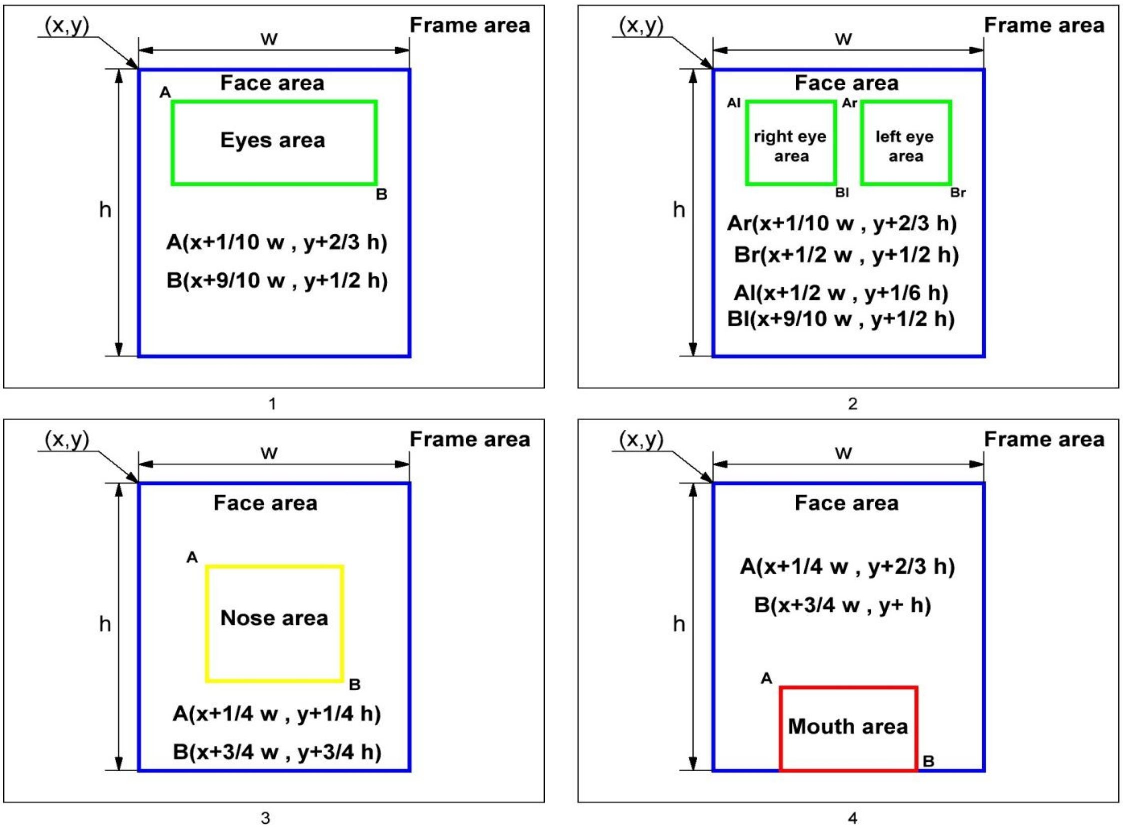


Fig. 2. Zones for searching faces and their elements in the frame (coordinates of individual search zones)

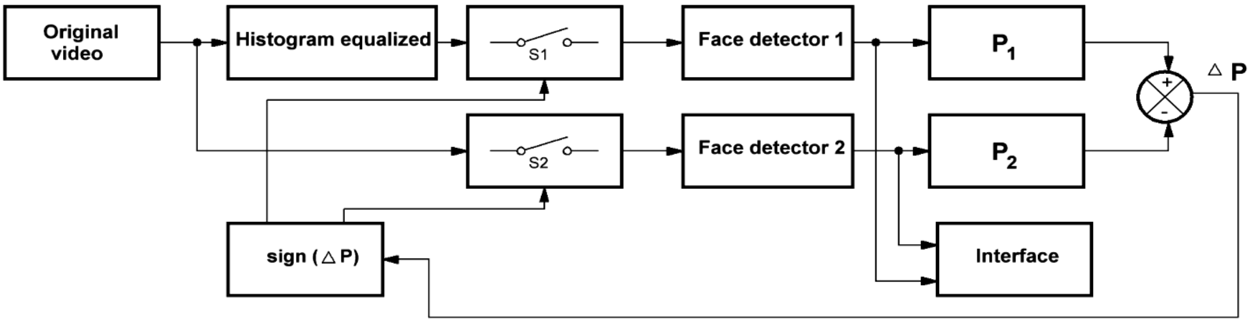


Fig. 3. Generalized structure of the adaptive algorithm for detecting faces

According to the authors, this method of adaptation to changes in scene illumination is much better than the method for estimating the average brightness of the MB frame, since its use does not impose significant restrictions on the processing speed of video data.

Practical implementation of algorithms

Working versions of the face detection algorithms were created taking into account the approaches and resources described above. The task was to find faces, eyes, nose and mouth. In this case, eye detection was carried out in two ways - using a classifier to detect the eyes in the general search area and the classifiers of separate detection (the right and left eyes separately). Next, compare the two methods in terms of the probability of eye detection.

To save resources and simplify the implementation of algorithms, our project used a set of pre-trained Haar cascading classifiers for the corresponding facial elements imported from the OpenCV library. It is worth recalling the need to place these classifiers in the root folder of Python. This helps to avoid errors in finding ways to access them and speed up the processing of the original data. The set of these classifiers is the next:

- face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
- eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')
- right_eye_cascade = cv2.CascadeClassifier('haarcascade_righteye_2splits.xml')
- left_eye_cascade = cv2.CascadeClassifier('haarcascade_lefteye_2splits.xml')
- nose_cascade = cv2.CascadeClassifier('haarcascade_nose.xml')
- mouth_cascade = cv2.CascadeClassifier('haarcascade_mouth.xml')

Since the methods and tools for creating algorithms have already been discussed in sufficient detail, we will dwell only on the structural features of constructing new algorithms.

Let's consider two different variants of constructing adaptive algorithms for detecting faces and their fragments on video sequences and assess the advantages and disadvantages of each of them. The structural diagrams of these algorithms are shown in Fig. 4, 5. In the first version, the principle of adaptation to changes in the level of scene illumination is completely implemented, based on determining the values of the current difference in the probabilities of correct detection of faces and their elements when using the frame equalization procedure and without it (see Fig. 4). The main disadvantage of this method is a noticeable decrease in performance, since in this situation the amount of computational operations is actually doubled. The one shown in Fig. 5, the variant of constructing an adaptive algorithm for searching faces in video frames is much simpler, since the procedure for equalizing video data is only used in those situations when, due to poor lighting conditions in the current frame, the face detection did not take place, despite its presence in the frame.

This approach is much more economical - the number of calculations is almost half that of the first version of the algorithm construction. This allows to confidently carry out processing of video data in real-time.

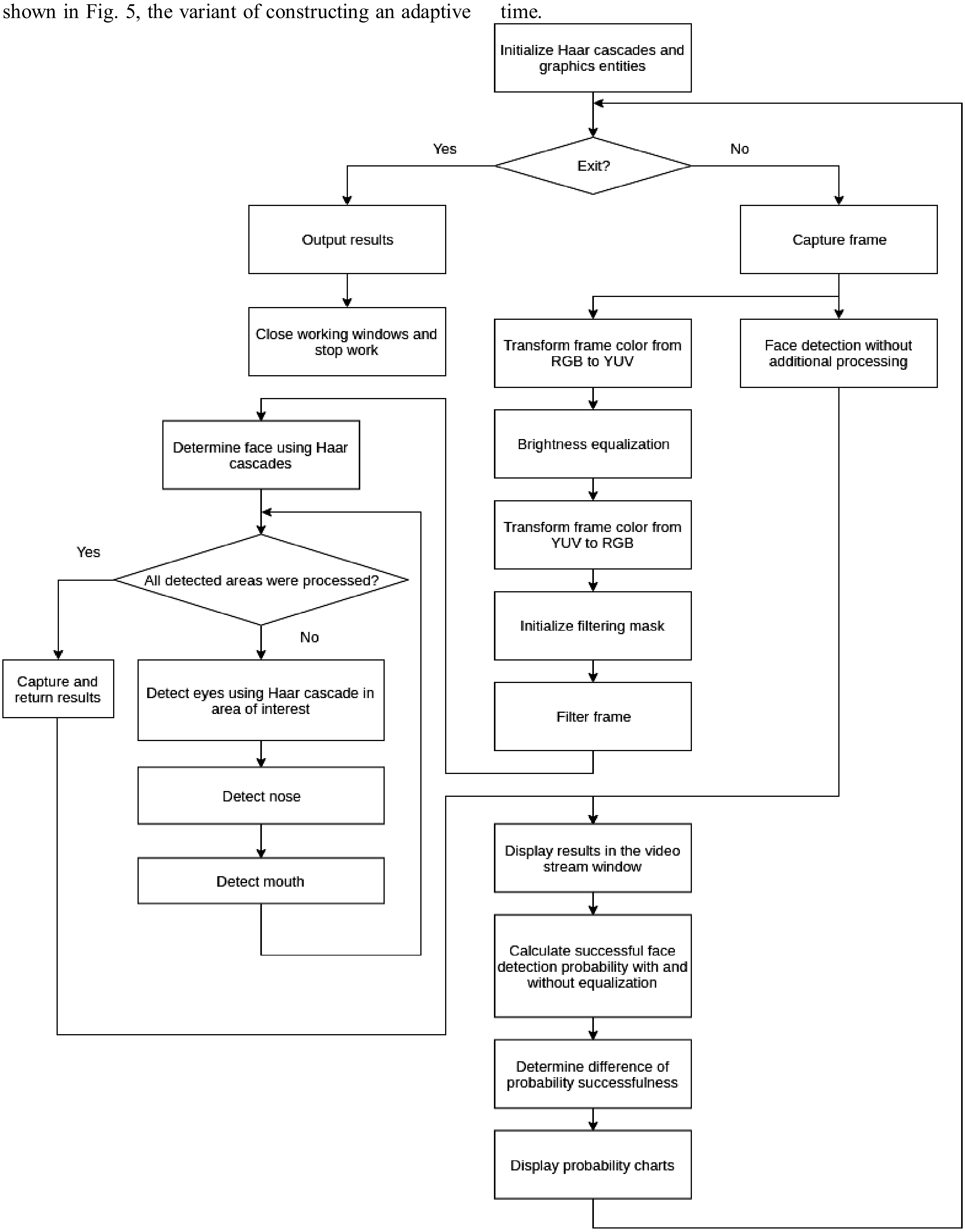


Fig. 4. Adaptation of the algorithm to lighting conditions according to the probability of correct detection of faces in the frame

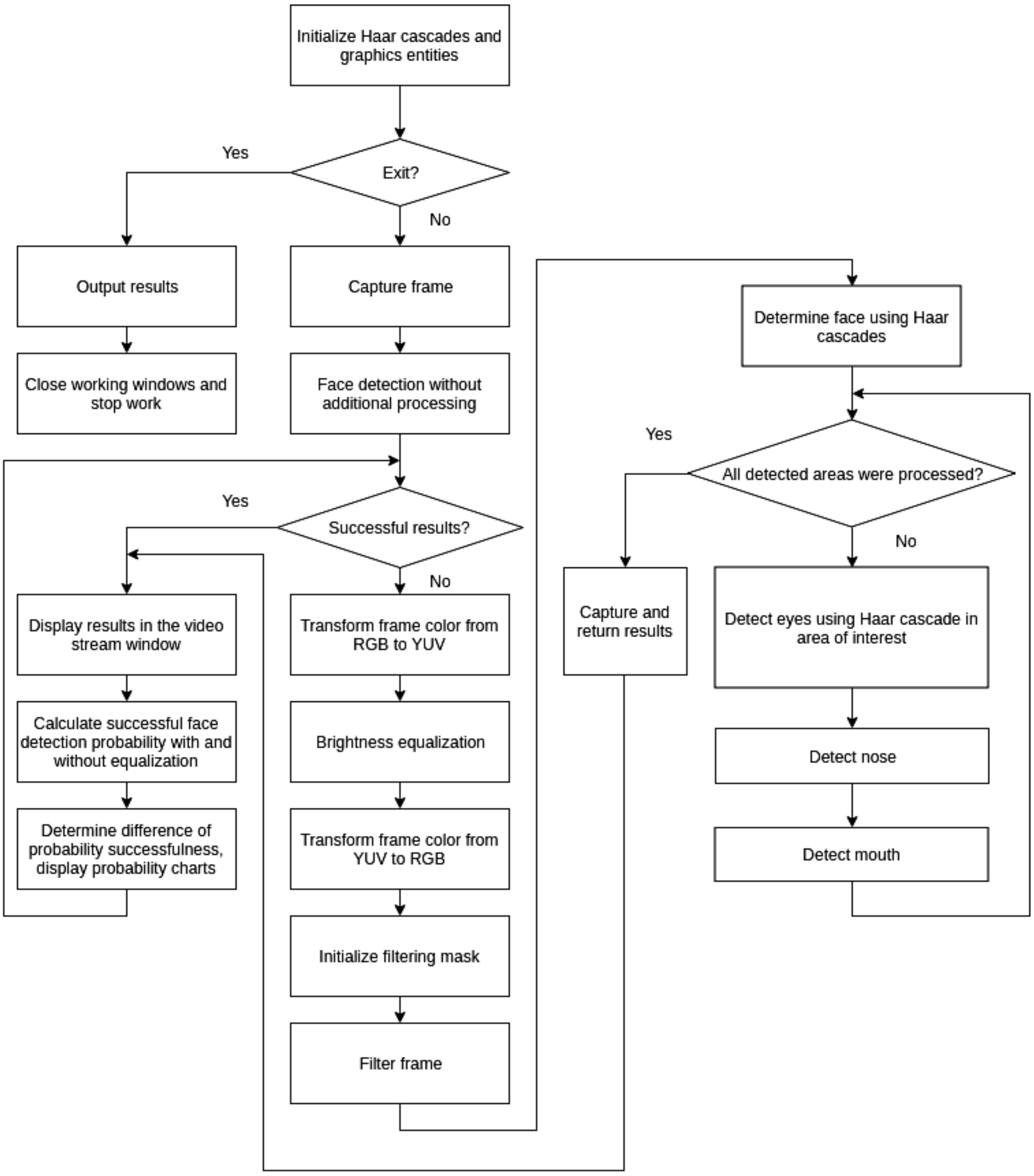
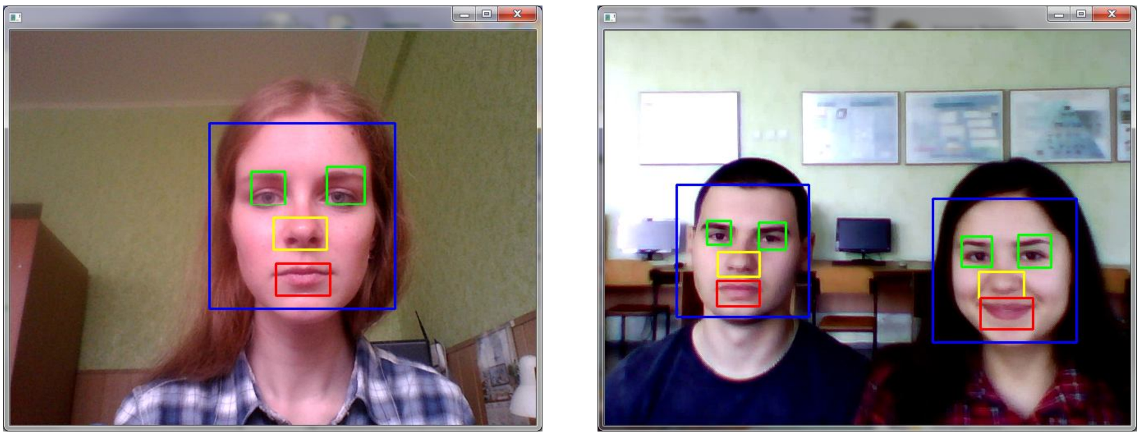


Fig. 5. A simplified version of the algorithm adaptation for detecting faces to lighting conditions

The results of detection of one or more persons using the proposed algorithms are shown in Fig. 6. It should be noted that the most stable results of the face and its elements detection are observed under the condition of the frontal location of the light source of the scene. A favorable geometric factor of the detection procedure assumes the head inclinations are not more than 30^0.

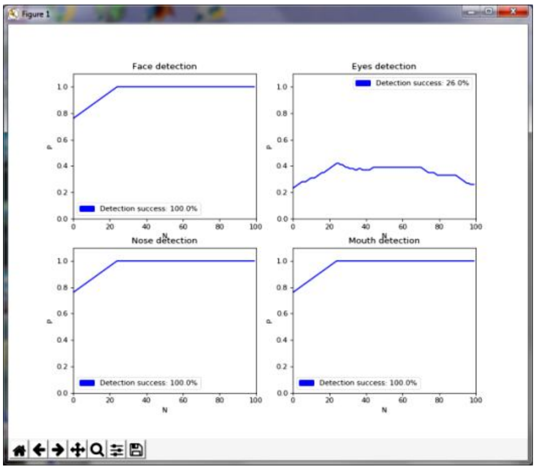


a b

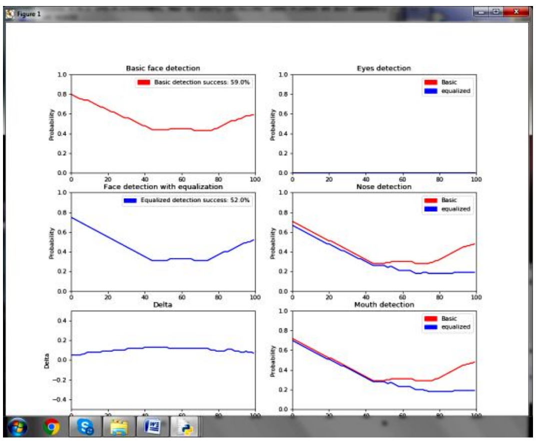
Fig. 6. Results of detection of faces and their elements in the frame

Results of experimental studies

Specific properties of algorithms for detecting faces based on the use of trained cascading classifiers determine the probabilistic nature of their functioning. Therefore, the main indicator of the quality of such algorithms in our work is considered to be the probability of a correct detection of the face and its elements. For clarity of the results obtained, in the first variant of the detection algorithm construction, a special graphic form was synthesized, inside of which, in individual windows, the probability of detecting faces during work without using the equalization procedure (red curves) and using it (blue curves) for the face and its elements (eyes, nose and mouth). In addition, in the lower left window of this form, the current difference in the probabilities of correct detection is displayed (when the video frame uses equalization and does not). The form of this generalized reporting form is clearly shown in Fig. 7, a. Similarly, a graphical form was constructed for the current estimation of detection efficiency and for the second algorithm. It contains four windows that display the current probabilities of correct detection of the face, eyes, nose and mouth (see Fig. 7, b). These means of visualization of current detection processes are quite effective, but they do not answer the main question - what is the efficiency of the algorithm in the long analysis of the video sequence?



a



b

Fig. 7. Windows of the current probability control of the correct detection of faces and their elements

For a generalized evaluation of the efficiency of algorithms, two methods are possible: qualitative (visual) and quantitative. In the first case, a visual observation is made of the multicolored rectangles that limit the detected face and its elements (eyes, nose and mouth). If the process of their drawing is perceived as continuous, then the quality of detection can be considered acceptable (equal to ~ 100 %). However, the inertia of the viewer's view at a high frame rate (30/s) does not allow visual perception of the missing frames for detecting individual frames. This can significantly distort the results of testing. Therefore, for quantitative evaluation of the effectiveness in the program, a one-dimensional array is formed for all frames of the video sequence (1 for the fact of detection of the face in the frame, and 0 for the case of missing the face), along which probabilistic characteristics of the algorithm's performance are constructed. However, one should keep in mind that for this, a reliable annotated test video sequence is needed. The simplest version of this sequence is the video sequence, which has a face for each detection in each frame, the lighting conditions (front light source) are met and the geometric factor is maintained (the head inclination is less than 30^0). Of course, you can use a more complex test video, which at certain intervals of time, the faces in the frame are missing. Then it becomes possible to evaluate not only the probability of correct detection of persons, but also the probability of false detection (detection of a person, provided it is not in the frame).

There is a special form of a generalized evaluation of the effectiveness of the algorithm for detecting faces and their main elements for a video sequence of 1000 frames is shown in Fig. 8. The duration of such a recording is ~ 33 s. For the detection procedures of all elements, periodograms are constructed, where for each of the N frames, 1 (if detected successfully) is placed in correspondence and 0 (if not detected) of the corresponding face element. In addition, for each periodogram, the probability of correct detection on a sample of 1000 frames is calculated. This form allows us to obtain not only generalized estimates of the detection probabilities, but also clearly shows which frames the detection was not performed on.

To assess the effectiveness, a number of test video recordings that meet the specified requirements were made. Records were made in the format “.avi” with a frame size of 480x640 pixels from a fixed camera. Significant artifacts, due to human movement in the frame, were excluded. The distance from the face to the camera lens was ~ 1 m. To control the level of illumination of the scene, when recording test video recordings, was used a Yu-16 lux meter with a photocell F102, which allows to fix the level of illumination with accuracy ±10% .

Since the proposed algorithms should automatically stabilize the brightness of the frames of the analyzed video sequence irrespective of the level of illumination, first of all the results of the detection of persons in records with different levels of illumination of the scene were compared. The results of this experiment are summarized in the Table 1. Analysis of tabular data showed that the probability of correct detection of individual facial elements increases with increasing level of illumination, provided that data processing is performed without automatically adjusting the average brightness of the frame. In cases where the equalization procedure is used to adjust the brightness, the probability of detection remains stable. In other words, with a low level of illumination, the efficiency of algorithms with automatic brightness adjustment is higher, and when the scene is brightly lit, the quality of work of both types of algorithms is approximately the same. It is necessary to note the high quality of the proposed algorithms, since similar modern technical solutions provide the probability of correct detection at the level of 95 - 97%.

Fig. 8. Window for assessing the effectiveness of detection of a face and its main elements on video data

Table 1 – **Performance indicators of the proposed algorithms**

Scene illumination, Lx	Face detection, P %	Eyes detection, P %	Nose detection, P %	Mouth detection, P %
Without equalization				
100	100	96,5	98,1	97,2
150	100	97,2	98,5	97,5
200	100	97,8	98,7	97,7
With equalization				
100	100	97,1	98,6	97,4
150	100	97,2	98,6	97,45
200	100	97,1	98,62	97,4

In addition, a study was made of the efficiency of the algorithms, depending on the distance of the face to the digital video recorder.

The initial distance was 0.7 m, and the final distance was 2.5 m. The rest of the shooting conditions were similar to previous researches. Note that the face detection procedure was insensitive to increasing the distance between the face and the video camera. But if this distance is increased by more than 1.5 m, this results in the interruption of the detection procedure for the facial elements (eye, nose and mouth). This limitation is related to the relative decrease in the size of the face in the frame, and can be eliminated by adaptively managing its dimensions.

Conclusion

New algorithms for detecting and detecting faces on digital images and video sequences are proposed. They implemented the ability to adaptively adjust the average brightness level of the frame, which allows to improve the basic performance. Probability of correct detection of faces for such algorithms is ~ 96%. They are implemented in Python programming language using OpenCV library resources.

This allowed real-time data processing. Using these results makes it possible to improve the quality of work and the reliability of results when recognizing faces in different systems.

REFERENCES

1. Viola, P. and Jones, M.J. (2001), "Rapid object detection using a boosted cascade of simple features", *IEEE Conf. on Computer Vision and Pattern Recognition*, Kauai, Hawaii, USA, Vol. 1, pp. 511–518.
2. Viola, P. and Jones, M.J. (2001), "Robust real_time face detection", *International Journal of Computer Vision*, Vol. 57, No. 2, pp. 137–154.
3. Joseph, Howse and Joe, Minichino (2015), *Learning OpenCV 3 Computer Vision with Python : Second Edition*, Packt Publishing, September, ISBN : 978-1-78528-977-4.

4. Saurabh, Kapur (2017), *Computer Vision with Python 3*, Packt Publishing, August 2017, ISBN: 978-1-78829-976-3.
5. Prateek, Joshi (2015), *OpenCV with Python By Example*, Packt Publishing, September 2015, ISBN: 978-1-78528-393-2.
6. Open Source Computer Vision, available at : http://docs.opencv.org/trunk (last accessed March 7, 2018).
7. Bovyrin, A., Druzhkov, P. and Erukhimov, V. (2016), Development of multimedia applications using libraries OpenCV and IPP, Intuit, Moscow, 382 p.
8. Comasch, F. Stuijk, S. Basten, T. and Corporaal H. (2013), "Rasw: A runtime adaptive sliding window to improve viola-jones object detection", *Distributed Smart Cameras* (ICDSC), 2013 Seventh International Conference.

Received (Надійшла) 11.06.2018
Accepted for publication (Прийнята до друку) 15.08.2018

Адаптивні алгоритми детектування обличчя
і оцінка ефективності їх використання

К. Ю. Дергачов, Л. О. Краснов, О. О. Челядін, А. Я. Зимовін

Предмет дослідження - виявлення і розпізнавання осіб. **Метою цієї статті** є створення модифікованих алгоритмів розпізнавання осіб, які забезпечують автоматичну стабілізацію яскравості аналізованого зображення незалежно від рівня яскравості. Запропоновано методику оцінки ефективності їх роботи в порівнянні з класичним алгоритмом. **Методи дослідження.** Найпопулярніший метод, який використовується для пошуку області обличчя на зображенні, - це метод Віола-Джонса, який популярний завдяки своїй високій швидкості і ефективності. Він заснований на поданні цілісного зображення, на методі побудови класифікаторів на основі адаптивного алгоритму прискорення (AdaBoost) і на комбінаційних класифікаторах в каскадному структурному методі. Метод Віола-Джонс по-перше використовує каскади вейвлетов (примітивів) - функції Хаара, що представляють розбивку заданої прямокутної області на безлічі різних типів прямокутних субрегіонів. Все це дозволило створити детектор особи, який працює в режимі реального часу з досить високою якістю. Однак існує безліч тривожних факторів, які обмежують ефективність роботи такого алгоритму. Основними з них є двозначність просторового положення особи на уже згадуваному зображенні і низька якість освітлення сцени. **Результати дослідження.** Запропоновано адаптивні алгоритми виявлення та детектування облич на цифрових зображеннях і відеопослідовністях в режимі реального часу, засновані на методі Віоли-Джонса. Для компенсації впливу змін рівня освітленості сцени на якість виявлення облич в класичну структуру алгоритму додатково введена процедура автоматичної стабілізації яскравості кадру. Описано структуру алгоритмів і розроблено програмне забезпечення для розпізнавання облич на мові Python з використанням ресурсів бібліотеки OpenCV. Обробка відеоданих здійснюється в реальному масштабі часу. Запропоновано і програмно реалізована оригінальна методика оцінки ефективності роботи алгоритмів за критерієм максимуму ймовірності правильного виявлення облич і їх головних елементів (очей, носа, рота). Зіставляються результати роботи класичного та запропонованих алгоритмів. Наводяться приклади роботи та результати тестування програмного забезпечення. **Висновок.** Використання отриманих результатів дозволяє поліпшити якість роботи і надійність результатів при розпізнаванні осіб в різних системах.

Ключові слова: виявлення і детектування обличчя; метод Віоли-Джонса; автоматична стабілізація яскравості кадру; ймовірність правильного виявлення осіб і їх головних елементів.

Адаптивные алгоритмы детектирования лиц
и оценка эффективности их использования

К. Ю. Дергачев, Л. А. Краснов, А. А. Челядин, А. Я. Зимовин

Предмет исследования - обнаружение и распознавание лиц. **Целью этой статьи** является создание модифицированных алгоритмов распознавания лиц, которые обеспечивают автоматическую стабилизацию яркости анализируемого изображения независимо от уровня яркости. Предложена методика оценки эффективности их работы по сравнению с классическим алгоритмом. **Методы исследования**. Самый популярный метод, используемый для поиска области лица на изображении, - это метод Виола-Джонса, который популярен благодаря своей высокой скорости и эффективности. Он основан на представлении целостного изображения, на методе построения классификаторов на основе адаптивного алгоритма ускорения (AdaBoost) и на комбинационных классификаторах в каскадном структурном методе. Метод Виолы-Джонса во-первых использует каскады вейвлетов (примитивов) - функции Хаара, представляющие разбивку заданной прямоугольной области на множества различных типов прямоугольных субрегионов. Все это позволило создать детектор лица, который работает в режиме реального времени с довольно высоким качеством. Однако существует множество тревожных факторов, которые ограничивают эффективность работы такого алгоритма. Основными из них являются двусмысленность пространственного положения лица на анализируемом изображении и низкое качество освещения сцены. **Результаты исследования.** Предложены адаптивные алгоритмы обнаружения и детектирования лиц на цифровых изображениях и видеопоследовательностях в режиме реального времени, основанный на методе Виолы-Джонса. Для компенсации влияния изменений уровня освещенности сцены на качество обнаружения лиц в классическую структуру алгоритма дополнительно введена процедура автоматической стабилизации яркости кадра. Описана структура алгоритма и разработано программное обеспечение для распознавания лиц на языке Python с использованием ресурсов библиотеки OpenCV. Обработка видеоданных осуществляется в реальном масштабе времени. Предложена и программно реализована оригинальная методика оценки эффективности работы алгоритмов по критерию максимума вероятности правильного обнаружения лиц и их главных элементов (глаз, носа, рта). Сопоставляются результаты работы классического и предложенных алгоритмов. Приводятся примеры работы и результаты тестирования программного обеспечения. **Вывод.** Использование полученных результатов позволяет улучшить качество работы и надежность результатов при распознавании лиц в разных системах.

Ключевые слова: обнаружение и детектирование лица; метод Виолы-Джонса; автоматическая стабилизация яркости кадра; вероятность правильного обнаружения лиц и их главных элементов.