Implementing Open Source Biometric Face Authentication for Multi-factor Authentication Procedures

Natalya Minakova and Alexander Mansurov(✉)

Altai State University, Lenin Avenue 61, Barnaul 656049, Russia
minakova@asu.ru

Abstract. This study proposes a solution that extends the capabilities of web information systems with single-factor authentication by introducing an additional authentication factor based on biometric face recognition. The proposed solution design and its main operation steps are presented and discussed. The solution utilizes the standard multimedia functionality of popular web browsers and supports available or built-in image capturing devices (photo and web cameras). Robust program algorithms from the open source computer vision library are used for face image processing and analysis. Experimental testing and validation of the algorithms for face localization and recognition are conducted with image sets produced with consideration of reality. Experimental results demonstrate high effectiveness with a success rate of 80% … 93% for the solution based on the local binary pattern face localization algorithm with the local binary pattern histogram face recognition algorithm.

Keywords: Biometrics · Face authentication · Face recognition · Computer vision · Local binary pattern

1 Introduction

Modern web applications and web information systems are complex and sophisticated solutions that typically process information with different levels of confidentiality. Such solutions often become targets of various hacker attacks when the attackers try to obtain the data circulated in web information systems by exploiting the weaknesses of web solution algorithms or bugs in software. According to sources [1–3], faults related to the authentication stage (or "Broken Authentication" in [1]) are rated as the second among the top 10 web application security risks, with up to 45% of similar vulnerabilities found during the conducted analysis [2]. Besides, there are several essential moments mentioned in [1] as crucial, like usage of well-known passwords, weak password database protection, susceptibility to brute-force attacks, and (what is important) missing or non-effective multi-factor authentication.

Using multi-factor authentication and adopting additional authentication factors can indeed increase the efficiency of the identification and authentication stage of web applications and web information systems [4, 5]. The research report "The State of Strong

V. Jordan et al. (Eds.): HPCST 2021, CCIS 1526, pp. 371–379, 2022.
https://doi.org/10.1007/978-3-030-94141-3_29

Authentication 2019" produced by Javelin Strategy and Research [3] states that there are many widely used supplementary authentication factors (that follow the first traditional password factor), such as pre-generated one-time passwords, SMS passwords, hardware cryptographic keys, and security questions. At the same time, biometric information inextricably linked with the web system user should also be considered the reliable authentication factor. Nowadays, fingerprint scanning (44%) and face recognition (10%) are actively used due to the widely spread mobile platforms already equipped with the necessary scanners [3, 6].

Biometric authentication requires certain cooperation on the user side and hardware and software support to acquire the user's biometric data. However, modern smartphones, laptops, tablet computers, and workstations are capable of obtaining various biometric data easily with minimal cooperation of the user. For example, behaviometric data can be collected silently, and the voice and face image of a user can be acquired with a built-in microphone or camera and simple instructions. The latter (face image, in particular) appears to have the potential for being used in biometric solutions and systems.

Not all current biometric face identification and recognition solutions can be considered open source and free to use [7]. It is also noted that their functionalities differ significantly. Some solutions are inclined to deprecated methods of utilizing multimedia devices and require additional supporting software components to be downloaded or installed on the user side.

Therefore, it is of importance to propose a solution for web information systems with single-factor authentication. The solution should introduce an additional authentication factor based on the biometric face recognition and be open source. Several requirements should also be considered, such as the capability to be incorporated into the existing web system authentication procedure, support of the standard web browser multimedia functionality, and usage of available (or built-in) image capturing devices (photo and web cameras).

2 Solution Design and Software Implementation

2.1 Concept of the Solution

The proposed solution includes all the necessary steps and modifications for the user side and server side of the web information system. The overall design is shown in Fig. 1.

The solution can be incorporated into the web information system algorithms right after the first authentication step. The starting point of the proposed solution, in general, does not require any specific parameters for its operation. However, the solution should comply with the already implemented web information system application program interface (API). If several parameters inherited from the previous step have to pass through to the web system operation step after authentication, then they should be collected and passed at the exit point of the solution. Thus, compatibility with the web system API is achieved.

The proposed solution starts the pre-arranged HTML code on the user side to initiate the biometric authentication step. The HTML code contains verbal instructions for users and program instructions to activate available multimedia devices on the user side for face image capturing (typically, a web camera or frontal photo camera). The web browser on

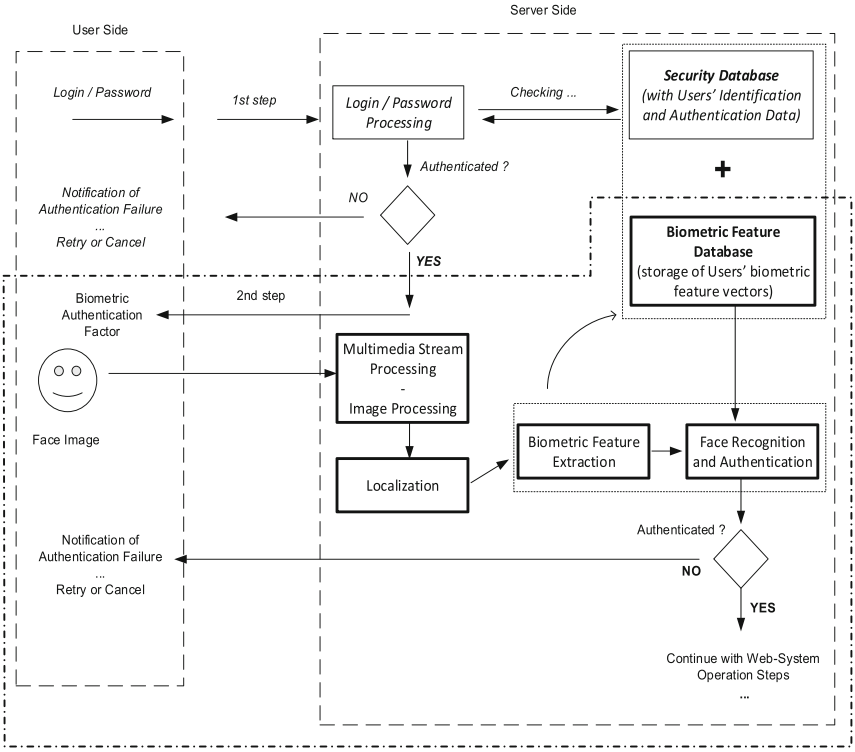

Fig. 1. The proposed solution overall design and its structure elements.

the user side fully handles all operations with the multimedia devices. The introduction of the HTML5 standard enhanced web browser multimedia capabilities significantly [8, 9]. Support of video streaming, capturing images from video streams, and operating the built-in multimedia devices on mobile platforms are of particular interest. WebRTC (Web Real-Time Communication) is a technology that enables Web applications and sites to capture and optionally stream audio and/or video media, as well as to exchange arbitrary data between browsers without requiring an intermediary [9]. Access to the data from photo and web cameras is provided by the *getUserMedia()* function that is supported by almost all latest web browsers on a variety of platforms running various operating systems [10] (Fig. 2).

Parameters of the *getUserMedia()* function can specify the data source and the desired format and quality of the video stream data. The JavaScript language is used to handle the operations. Program instructions necessary for engaging the camera and transmitting the acquired face image to the server side are incorporated into the web page designed for the biometric authentication step on the user side.

The captured face image is transmitted to the server side for processing and analysis. The server side algorithms can be developed separately and assembled into a complete project later [11]. Source codes of the algorithms can be produced independently or taken from the appropriate software library.

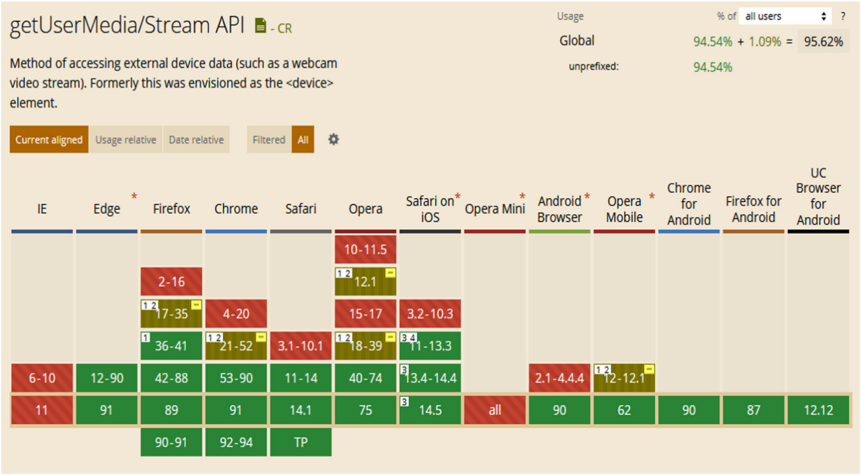

Fig. 2. Support of getUserMedia/Stream API by web browsers [10].

2.2 Processing Algorithms and Methods

Comprehensive studies [7, 12] suggest that the open source computer vision library OpenCV [13] and its PHP frontend PHP-OpenCV seem to be among the most effective open source software libraries for video and image processing and analysis. OpenCV includes several hundreds of computer vision algorithms, and there are many adaptive (trainable) algorithms based on artificial neural networks (ANN). These algorithms are highly effective for processing images, calculating specific features (feature vectors), localizing and detecting objects in images. Therefore, it seems prospective to utilize the algorithms included in the OpenCV software library.

In this study, the performance of several face localization methods is tested and evaluated to select the best performing method for its further implementation as the main one. The following face localization algorithms available in the OpenCV library are utilized: linear binary patterns (LBP) [14], FacemarkLBF [15], Haar cascades [16]. This part is necessary due to the importance of successful face localization. Once the face is localized in an image, it is possible to continue with face recognition. Otherwise, appropriate actions should be taken to acquire a new face image.

Face recognition is conducted using the trainable classifier based on the local binary pattern histograms (LBPH) [14, 17–19]. The LBP operator [14] is applied to the localized face image to calculate feature histograms representing local texture and shapes over the processed areas. The localized face image is divided into small regions from which LBP histograms are extracted to produce the resulting feature histogram vector (feature vector) (Fig. 3) [19]. Calculated feature vectors for successful classifications are stored in the Biometric feature database (when the user face image is processed for the first time) for further accumulation and enrichment of the training dataset.

The transmitted image undergoes normalization of brightness and contrast before being passed to the localization step. Unsuccessful localization of a face in the image or localization of the facial profile leads to disqualification of the image under processing.

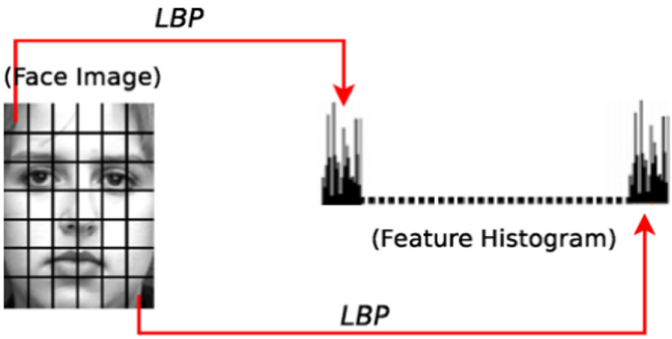

Fig. 3. Calculation of the feature vector using the LBPH method [19].

In this case, all operations start from the beginning, and the user side should provide another face image. The overall flowchart of face image processing and analysis is shown in Fig. 4.

3 Testing and Evaluation

Several available algorithms are taken for testing and evaluation of their performance to select the most efficient ones. These algorithms provide face localization and detection in processed images which can be acquired in various environments and be of different quality.

Testing and performance evaluation of face localization algorithms is conducted using the assembled set of 514 images of various faces. The assembled set contains images from several groups, and each group has almost the same number of images. These groups are named according to the specific 'trait' that all images within the group possess. The assembled set includes images from the following groups:

- ideal images with a proper light balance and faces positioned at the center ("*Ideal*");
- images with faces out of focus or images contain parts of faces only ("*Part-Face*");
- images with partially covered faces ("*Hidden*");
- images with excessive darkness or images with shadows on faces ("*Dark*");
- images with excessive light ("*Light*").

Experimental results are shown in Table 1 with percentages that represent successful face localization for each group.

"Haarcascade alt" and "*Haarcascade alt2"* algorithms demonstrate good results with clear tendencies to localize small faces or embossed objects. These algorithms provide the best results when processing images with excessive light or images from the "*Part-Face*" group. The algorithms are suitable for processing images acquired using mobile platforms. The "*Haar cascade alt tree"* algorithm shows good performance when dealing with images with partially covered faces or with faces out of focus.

The "*LBF"* algorithm is able to detect facial key points and fit for detailed localization. However, "*LBP"* and "*LBP improved"* algorithms demonstrate accurate and stable

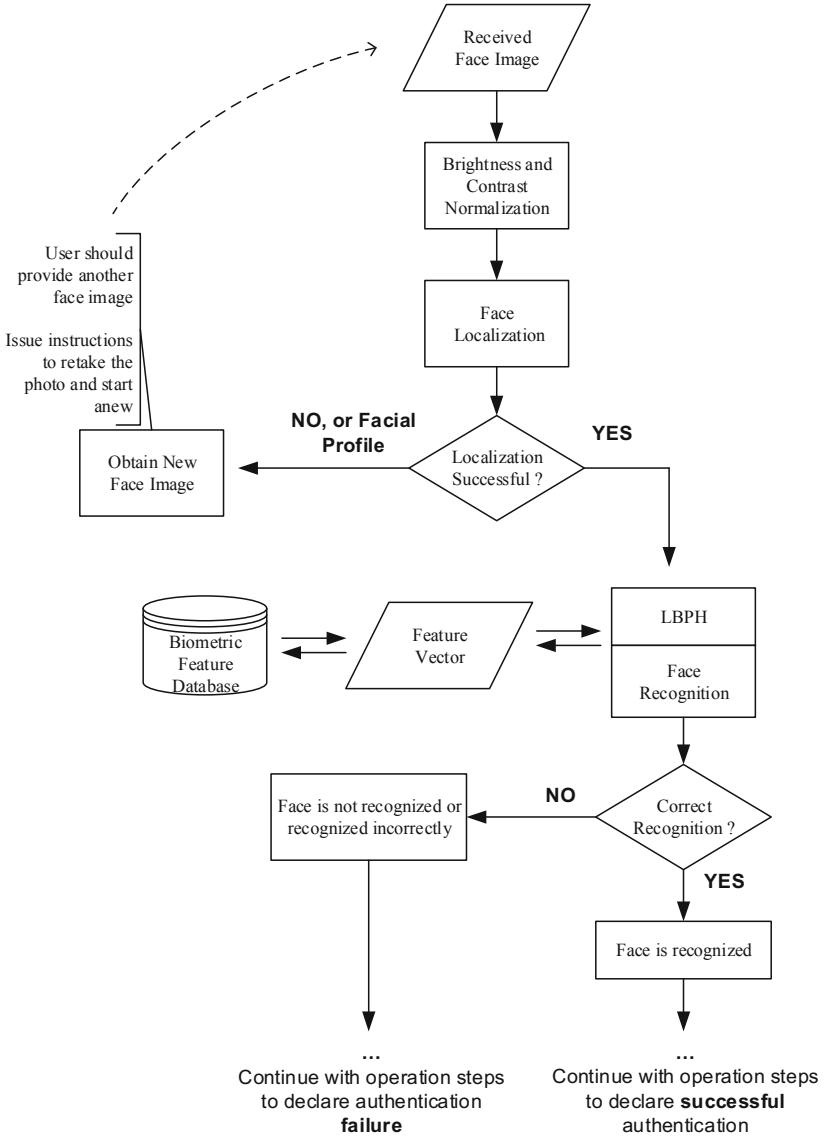

Fig. 4. Flowchart of face image processing and analysis.

Table 1. Experimental results of face localization for several groups of face images.

Algorithm\Image group	Dark	Hidden	Ideal	Light	Part-Face
Haarcascadealt	0.44	0.48	0.97	0.94	0.84
Haarcascade alt2	0.59	0.43	0.93	0.93	0.75
Haarcascadealttree	0.14	0.93	0.29	0.24	0.96
Haarcascadedefault	0.39	0.29	0.82	0.80	0.74
LBF	0.59	0.59	0.90	0.68	0.66
LBP	0.46	0.61	0.84	0.70	0.73
LBP improved	0.74	0.53	0.94	0.94	0.64

results for all groups of images in the assembled set. Thus, the "*LBP improved"* algorithm is selected as the best performing algorithm for its implementation as the main localization algorithm of the proposed solution.

The LBPH-based trainable classifier is used further to perform face recognition. In this study, the LBPH based classifier produces two possible outputs – "affirmative" (recognition is done, even if the face is matched incorrectly) and "negative" (face is not recognized / no match found).

There are two experimental samples prepared to train and evaluate the LBPH based classifier. Test sample "A" contains 58 various face images of the same person taken under various conditions. These conditions are equivalent to the ones for the image groups of the previously discussed assembled image set for localization testing. The training sample includes 20 images from the sample "A" and 100 images of other (different) persons.

Test sample "B" contains 50 various high-quality face images of the same person. There are no images with partially covered faces or images with faces out of focus in this test sample. However, images with excessive light or darkness are included in the test sample "B". The training sample for this case includes 20 images from the sample "B" and 100 images of other (different) persons. All images comply with the conditions mentioned earlier.

Table 2 demonstrates the performance results (in percentage) of the trainable LBPH-based classifier with the two experimental samples.

Table 2. Performance results of the LBPH-based classifier.

Results\Samples	Test sample "A"	Test sample "B"
Correct recognition	0.7	0.87
1st type error	0.1	0.06
2nd type error	0.2	0.07

Here, the "*1st type error*" stands for erroneous recognition of face images belonging to one user as belonging to another. The "*2nd type error*" stands for unsuccessful recognition when the face is not matched with the face of any user.

Results of the experiment demonstrate a high rate of correct face recognition. It can be seen that the most optimal case would be direct utilization of two output classes and adoption of the "1st type error" results as successful face recognition. However, the biometric authentication factor, in this case, should be interpreted as a very soft authentication step with only two outcomes:

1. the face is matched, and so the user is known to the web information system;
2. the face is not matched, and so the user is unknown to the web information system.

The rate of correct face recognition can be increased by using more robust adaptive algorithms and models [17]. Also, the proposed biometric authentication step can be improved by utilizing additional biometric data and processing algorithms. For example, face recognition can be supplemented by iris scanning [20] or any other biometric technique.

After the biometric authentication step is acknowledged as successful, the control returns to the web information system algorithms at the point located right after the initial authentication procedure of the web information system.

4 Conclusion

This paper presents the solution that extends the capabilities of web information systems with single-factor authentication by introducing an additional authentication factor based on biometric face recognition. The proposed solution is based on algorithms from the open source software library. It uses the standard web browser multimedia functionality and available (or built-in) image capturing devices (photo and web cameras).

The proposed solution can be easily integrated into existing authentication procedures of web information systems. Face localization and recognition algorithms are based on adaptive ANN techniques and demonstrate a high success rate. They can be replaced during further operation by more robust and prospective ones to fully satisfy the specific requirements of a particular web information system and incorporate all achievements in the area of biometric data analysis and face recognition.

References

1. The OWASP: OWASP Top Ten Web Application Security Risks. https://owasp.org/www-project-top-ten/, Accessed 14 May 2021
2. Positive Technologies: Web Applications Vulnerabilities and Threats: Statistics for 2019. https://www.ptsecurity.com/ww-en/analytics/web-vulnerabilities-2020/, Accessed 14 May 2021
3. Pascual, A., Maarchini, K.: The State of strong authentication 2019. Adoption Rises under New Threats and Regulations. Report, Javelin Strategy & Research (GA Javelin LLC), Pleasanton, CA, USA (2019)

4. Shah, Y., Choyi, V., Subramanian, L.: Multi-factor authentication as a service. In: IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud 2015), San-Francisco, USA, pp. 144–150. (2015). https://doi.org/10.1109/MobileCloud.2015.35
5. Jacomme, C., Kremer, S.: An extensive formal analysis of multi-factor authentication protocols. In: 2018 IEEE 31st Computer Security Foundations Symposium (CSF) 2018, Oxford, UK, 2018, pp. 1–15. (2018). https://doi.org/10.1109/CSF.2018.00008
6. Ometov, A., Bezzateev, S., Mäkitalo, N., Andreev, S., Mikkonen, T., Koucheryavy, Y.: Authentication: a survey. Cryptography **2** (2018). https://doi.org/10.3390/cryptography2010001
7. Masek, P., Thulin, M.: Evaluation of face recognition APIs and libraries. https://gupea.ub.gu.se/bitstream/2077/38856/1/gupea_2077_38856_1.pdf, Accessed 14 May 2021
8. The W3C: HTML Media Capture. https://www.w3.org/TR/html-media-capture/, Accessed 14 May 2021
9. MDN Web Docs: WebRTC API. https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API, Accessed 14 May 2021
10. The "Can I use" Project: "getusermedia". https://caniuse.com/?search=getusermedia, Accessed 14 May 2021
11. Minakova, N., Petrov, I.: Modeling and prototyping of biometric systems using dataflow programming. J. Phys. Conf. Ser. **944**(1), 012080 (2018). https://doi.org/10.1088/1742-6596/944/1/012080
12. Ranjan, R., Bansai, A., Zheng, J., Xu, H., Gleason, J., Lu, B., et al.: A fast and accurate system for face detection, identification, and verification. IEEE Trans. Biometrics, Behav. Identity Sci. **1**(2), 82–96 (2019). https://doi.org/10.1109/TBIOM.2019.2908436
13. OpenCV: OpenCV Modules. https://docs.opencv.org, Accessed 14 May 2021
14. Ahonen, T., Hadid, A., Pietikäinen, M.: Face recognition with local binary patterns. In: Pajdla, T., Matas, J. (eds.) ECCV 2004. LNCS, vol. 3021, pp. 469–481. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24670-1_36
15. Ren, S., Cao, X., Wei, Y., Sun, J.: Face alignment via regressing local binary features. IEEE Trans. Image Process. **25**(3), 1233–1245 (2016). https://doi.org/10.1109/TIP.2016.2518867
16. Hapsari, D.T.P., Berliana, C.G., Winda, P., Soeleman, M.A.: Face detection using haar cascade in difference illumination. In: 2018 International Seminar on Application for Technology of Information and Communication, Semarang, Indonesia, 2018, pp. 555–559 (2018). https://doi.org/10.1109/ISEMANTIC.2018.8549752
17. Goncharov, V.: Tutorial for computer vision and machine learning in PHP 7/8 by opencv (installation + examples + documentation). https://github.com/php-opencv/php-opencv-examples, Accessed 14 May 2021
18. Deeba, F., Memon, H., Dharejo, F., Ahmed, A., Ghaffar, A.: LBPH-based enhanced real-time face recognition. Int. J Adv. Comput. Sci. Appl. **10**(5), 274–280 (2019). https://doi.org/10.14569/IJACSA.2019.0100535
19. Shan, C., Gong, S., McOwan, P.W.: Facial expression recognition based on local binary patterns: a comprehensive study. Image Vis. Comput. **27**(6), 803–816 (2009)
20. Minacova, N., Petrov, I.: Method of preliminary localization of the iris in biometric access control systems. IOP Conf. Ser. Mater. Sci. Eng. **93**, 012056 (2015). https://doi.org/10.1088/1757-899X/93/1/012056