Journal of
Control & Instrumentation

https://journals.stmjournals.com/joci

ISSN: 2229-6972 (Online)
ISSN: 2347-7237 (Print)
Volume 16, Issue 1, 2025
January–April
DOI (Journal): 10.37591/JoCI

Review JoCI

# Face and Spoofing Detection Via Genetic Algorithm-Based Feature Selection with MTCNN

Karan R. Talwalkar[1,*], Kshitj Navale[2], Aditya Ashok[3]

***Abstract***

*Face detection and liveness detection in various environments, such as different lighting effects, occlusion, complex backgrounds, and different poses and angles, play an important role in facial recognition or detection purposes. In this research, we propose an improved algorithm for face detection and liveness, spoofing detection via genetic algorithm for feature selection, MTCC detecting the edge of the facial image by Canny filter, and spotting the face from the image, and whether the image is spoofed or not. These experimental results used various types of complex image datasets and generated datasets. In this algorithm, edge detection is done by using the Canny detector. We tried to solve the problem of low-contrast images. In the preprocessing stage, we used an image-denoising algorithm to remove noise from the image. The detection rate has reached 100%. Real-time face detection using multitasking cascade convolutional neural network (MTCNN).*

**Keywords:** Face detection, expression detection, genetic algorithm, Canny detector, fast NÍ Means Denoising Colored MTCNN

## INTRODUCTION

ACE detection is a technique used to detect any number of faces in given images, videos, or facial detection biometric systems. Most human-computer interaction devices use facial detection as the preprocessing step and a recognition system for authentication or many other purposes. Currently, many face detection techniques are available. First, there are knowledge-based techniques that have certain guidelines regarding human facial traits, but have problems with their algorithms regarding different head positions [1, 2]. The second method uses features to identify people by observing their eyes or eyebrows. This technique often provides effective detection, but it also depends on variations in the head and the backdrop. It may also be unsuccessful if the image quality is inadequate [1, 2]. To overcome this restriction, we need new optimization methods that perform better on live databases and standards. For face detection, the following procedure is generally used: the image is first chosen from a standard database. Next, it was enhanced, which means contrast enhancement is one of the most important stages in face reorganization, and deduction will help to improve the face image before performing face recognition [3]. When comparing facial images to other methods that are known to exist, there are some differences. The proposed solution uses a process called contrast stretching or normalization, which modifies the brightness of the object in the image to provide clear visibility [4–6]. Second, we removed noise from the image to enhance the image quality through techniques such as filters, threshold value, and mean value, and the noise found in the image is removed by a denoising filter and median filter [7]. This Denoising filter is used in image processing to remove noise from images. This noise can be present in the image owing to various factors such as sensor limitations, transmission interference, or other environmental factors. Nonlocal means (NLM) is a popular

*Author for Correspondence
Karan R. Talwalkar
E-mail: karantalwalkar@gmail.com

[1-3]Student, Department of Electronics and Communication Engineering, Shri Vile Parle Kelavani Mandal Narsee Monjee Institute of Management Studies, Thane, Maharashtra, India

Receiving Date: December 20, 2024
Accepted Date: January 02, 2025
Published Date: January 07, 2025

**Citation:** Karan R. Talwalkar, Kshitj Navale, Aditya Ashok. Face and Spoofing Detection Via Genetic Algorithm-based Feature Selection with MTCNN. Journal of Control & Instrumentation. 2025; 16(1): 17–26p.

 

denoising algorithm that works by making use of the redundancy present in real photos. Rather than simply using the average of nearby pixels to eliminate noise, NLM computes the similarity between the two patches of the image by comparing them. The image was then denoised by applying a weighted average to these patches. Compared with conventional local averaging techniques, this method typically preserves more image information. Fast Algorithms for Denoising: Given the computational complexity of some denoising algorithms, there is ongoing research into developing faster variants or approximations that can achieve similar results with reduced computational costs. These fast algorithms often involve trade-offs between speed and denoising quality. The genetic algorithm (GA) is a population-based stochastic search technique that uses a stochastic search procedure based on a population to find nearly correct solutions to optimization and search problems. It was designed on the mechanisms of evolution and natural genetics; GA uses directed random searches to locate optimal solutions in multimodal landscapes and acts as an alternative to traditional optimization techniques. The main solution to this problem is the presence of chromosomes in the population. The series of populations for each upcoming and succeeding generation is designed by GA, and crossover and mutation are used as operators for the principal search mechanisms. The optimization of a given objective or fitness function is the main objective of the algorithm. Each perfect or effective solution for a chromosome is mapped using an encoding mechanism. Each chromosome that provides a satisfactory solution to the problem is evaluated using objective functions [8]. Finally, we can easily find the human face with the help of all these processes and can detect vulnerabilities, if any.

## LITERATURE REVIEW

Several studies have focused on enhancing the face recognition technology using various methodologies. A study [9] explored the effectiveness of a GA in comparison to Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) algorithms. This underscores the aim of improving recognition rates across different databases. Meanwhile, a study [10] discussed the synergy between GA and back-propagation neural networks (BPNN) in tackling face recognition challenges, emphasizing the significance of efficient face detection mechanisms in security systems. This combination harnesses the GA's evolutionary principles and BPNN's learning capabilities for pattern recognition tasks. In another vein, the study [11] introduced multitasking cascading convolutional networks for face recognition, highlighting its advantages, such as noncontact interaction and simplicity in equipment requirements. The proposed method utilizes a multitasking cascade convolutional neural network (MTCNN) comprising three convolutional networks (P-Net, R-Net, and O-Net) to enhance face detection performance while ensuring real-time processing capabilities. Testing on a dataset of 10,000 images, the system achieved a real-time recognition probability of 96.02% with a false recognition rate of less than 10%, outperforming the existing attendance recording systems.

Furthermore, a study [12] outlines a face recognition model tailored for real-time applications. It integrates FaceNet and Support Vector Machines (SVM) for classification, incorporating face detection using the MTCNN library and assessing scores using accuracy, precision, recall, and F1-score metrics. The system exhibited high accuracy rates across different head positions in front of the camera, achieving an average accuracy of 91% in live face recognition scenarios.

These studies collectively contribute to the ongoing advancements in face recognition technology, offering diverse methodologies and insights for improving the accuracy and efficiency of various applications [13].

## PROPOSED METHODOLOGY

In this proposed system, there are six stages of processing:

1. Image Acquisition
2. Image Enhancement
3. Preprocessing
4. Genetic Algorithm

 

5. Face Detection
6. Spoofing Detection

**Image Acquisition**

In this webcam, a direct image is provided by the dataset folder. If the input is selected by the webcam, it returns an image frame using the snapshot function obtained [14].

**Image Enhancement**

After the acquisition of the image from the webcam or directly from the dataset, image enhancement is necessary because of the various weather conditions, background complexity, and different lighting conditions. In this step, we increased the pixel value of the selected image by a constant value. In this process, some noise is introduced, so that we move to the preprocessing stage [15]. The flow Diagram is shown in Figure 1.

**Preprocessing**

The image must be resized to the required standard size after enhancing the image. Noisy pixels from an image are removed using a nonlocal means filter.

This class of filters is a nonlinear filter. It belongs to the class of denoising filters specifically targeted at the reduction or removal of noise from an image while attempting to preserve important image features such as edges and textures.

- Convert the input RGB image to grayscale to simplify processing [16].
- The grayscale image was resized to 256 × 256 pixels to standardize the input dimensions.
- Eliminate noise from the image using the cv2.fastNlMeansDenoisingColored function to enhance the quality of subsequent processing.
- The Canny edge detection algorithm is applied to the denoised image to extract the edges and emphasize the features relevant to face detection.

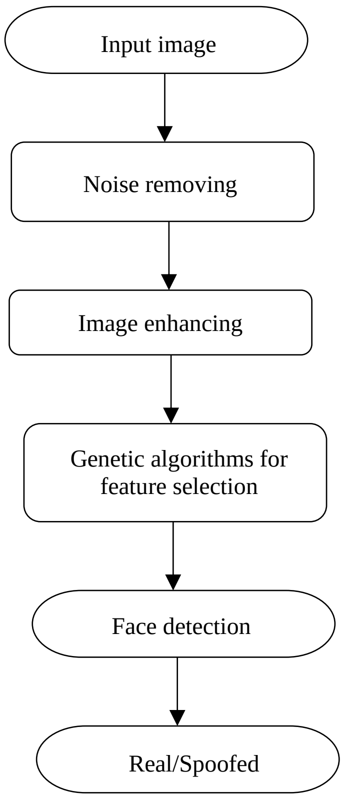


**Figure 1.** Flow diagram of the proposed method.

 

$$I(x_i, y_i) = \frac{\sum_j I(x_i, y_i), w_{i,j}}{\sum_j w_{ij}} \tag{1}$$

**Preprocessing**

The dataset was resized to a fixed aspect of the facial region (256 × 256) in the input image of this step. In some portions of the image, such as hair, the background increases noise for expression identification. In this step, features from the image are extracted by converting the image into grayscale into a binary format. Here, 1 represents the edge portion of the image and 0 represents the non-edge region. To identify the edges in the image, we used the Canny algorithm to convert it into a gray format. This converts a gray picture into a binary picture, so an analysis of each picture element is performed. The steps of preprocessing are shown in Figure 2, where 2(a) represents the input image and (b) is the binary image of the input [17].

**Genetic Algorithm**

Face detection using a genetic algorithm requires updating the population of the candidate solution (potential face detector) over generations to find the correct cluster of features to accurately detect faces in images. The Training Architecture of the model shown in Figure 3 provides an overview of how to achieve this.


**Figure 2.** (a) Preprocessing; (b) Edge detection.

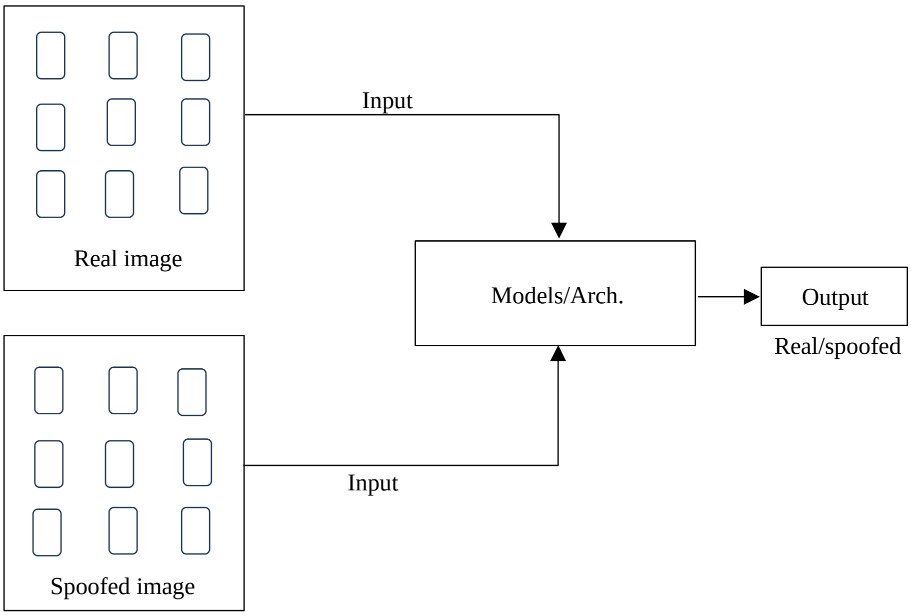


**Figure 3.** Training the architecture of the model.

***Representation***

represents the candidate solution. This could be a technique for identifying a particular face or set of features extracted from an image [18].

***Initialization***

Creating an initial population of candidate solutions or using some heuristics.

***Evaluation***

Evaluating each candidate's solution's fitness or performance using a fitness function, evaluating everyone based on their fitness score, which has the highest fitness score, will be taken further for offspring creation. The fitness function measures how well each solution detects the faces in each set of training images.

***Selection***

Select a subset of candidate solutions to form the next generation based on their fitness, who has great fitness scores and will be selected for further processing. Selection methods, such as tournament selection, roulette wheel selection, or elitism, are used to preserve the best solutions.

***Crossover***

Apply crossover or recombination to pairs of selected solutions to create offspring solutions. This mimicked the processes of reproduction and genetic recombination during biological evolution. For face detection, crossover can involve combining parameters or features from two-parent solutions.

***Mutation***

Random changes are introduced to the offspring solutions to maintain genetic diversity and explore new regions of the search space. It adds some randomness to the chromosomes of the individual to create genetic diversity and increase the probability of selection. Mutations can involve randomly modifying the parameters or features of a solution [19].

***Replacement***

The least-fit members of the population are replaced with new offspring solutions to form the next generation. The unfit test individual was terminated or replaced with an offspring. Several techniques are available for performing certain tasks.

***Termination***

Over several generations or until the process converges (i.e., there is no improvement in fitness).

***Testing***

The best solution was tested in a separate test to evaluate its performance. The use of GAs for face detection requires careful evaluation of parameters, selection of appropriate agents, and extensive testing to ensure performance. In addition, note that genetic methods may be good but are not guaranteed to find the best world and will get stuck in the right place. Therefore, multiple studies with different genes and test parameters are required to ensure robustness [20].

Process Used in this Proposed Genetic Algorithm

- Initialize the genetic algorithm parameters:
- Population size: 20 individuals.
- Number of generations: 15.
- Crossover probability: 0.8.
- Mutation probability: 0.2.
- Define the chromosome representation:

 

- Each chromosome consists of four elements representing the parameters of an ellipse:
  1. X0 (x-coordinate of ellipse center) - 8 bits.
  2. Y0 (y-coordinate of ellipse center) - 8 bits.
  3. a (semi-axis value) - 8 bits.
  4. b (major axis value) - 8 bits.
- Generate an initial population of ellipses with random parameter values.
- Evaluate the fitness of each chromosome:
- For each ellipse, the fitness was calculated by comparing the ellipse equation with the edge points extracted from the Canny image.
- Fitness is determined by the number of edge points that fall within the ellipse boundaries, considering a predefined tolerance for imperfect ellipses.
- Selection, crossover, and mutation operations are performed to evolve the population over multiple generations.
- Repeat the evolutionary process until reaching the specified number of generations.
- Bounding Ellipses in Rectangle Boxes:
- For each best-matching ellipse obtained from the genetic algorithm:
- Calculate the minimum bounding rectangle that encloses the ellipse.
- Represent the rectangle as a bounding box containing the potential face region.

**Face Detection with MTCNN**

- The rectangular regions are cropped from the original image based on the bounding boxes obtained in the previous step.
- Apply MTCNN to each cropped region to detect faces and facial landmarks.
- The MTCNN output was analyzed to determine the presence of genuine faces or potential spoofing attempts based on the detected facial features and their alignment.

***Real-Time Detection***

- The entire process is integrated into a real-time system capable of processing live video streams.
- Frames are continuously captured from the video stream, and preprocessing, ellipse matching, and face detection steps are applied in real-time [21, 22].
- Display the results indicating whether the detected faces were genuine or spoofed in the live video.

***Face Detection***

After image acquisition, image enhancement must be performed on the face image for low illumination, poor contrast, and different pose angles. The image visibility is enhanced by this, and the model can easily provide an accurate output with high accuracy. The structure of this mode is shown in Figure 4. We used a face cascade for face detection in the images [23].

## RESULTS

This proposed method was evaluated by using various images and facial expressions.

Accuracy = (No of Accurate faces detected) / (total number of images)

**Improved Face Spoofing Detection**

The integration of genetic algorithms with MTCNN and image preprocessing is expected to enhance the accuracy and robustness of face spoofing detection. By optimizing the parameters of ellipses to match faces in images, followed by precise face detection using an MTCNN, the algorithm can effectively differentiate between genuine and spoofed images. The dataset result of face detection is shown in Figure 5, and the Face Real/Spoof Output is shown in Figure 6.

 

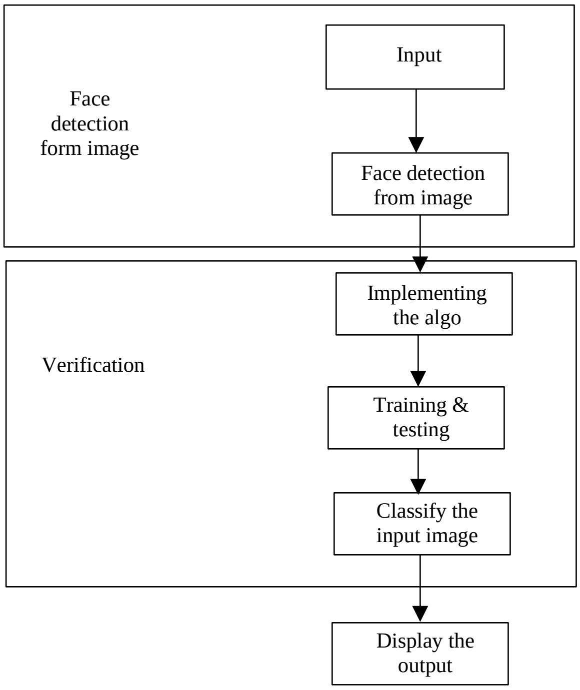


**Figure 4.** Structure of the mode.

**Figure 5.** Dataset result of face detection.

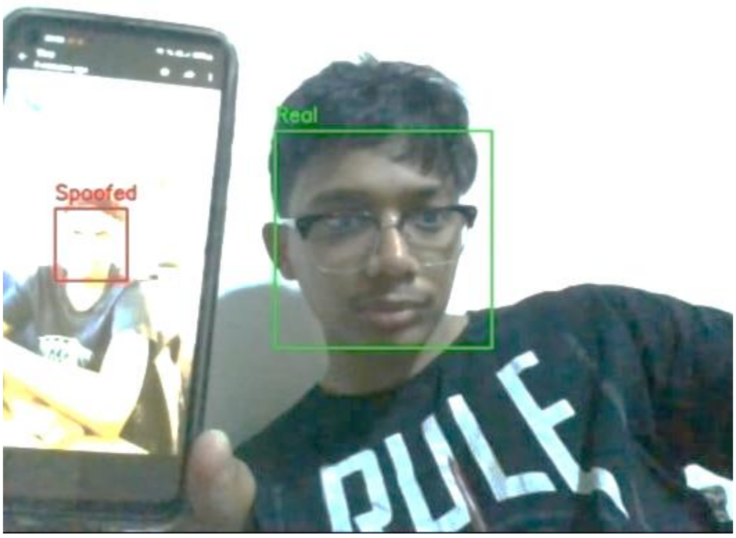


**Figure 6.** Face real/spoof output.

 

**Increased Efficiency**

The use of genetic algorithms allows for the adaptive optimization of ellipse parameters, potentially reducing the computational complexity of face detection. By focusing computational resources on the regions of interest identified by the genetic algorithm, the overall efficiency of the face recognition system is expected to improve. The Real Image Output is shown in Figure 7, and the Spoofed Image Output is shown in Figure 8.

**Reduced False Positives**

Preprocessing steps, such as noise elimination and edge detection, help refine the input images and reduce noise and irrelevant information. This, combined with the adaptive ellipse matching provided by genetic algorithms, is expected to result in fewer false positives in face and spoofing detection.

**Real-Time Detection**

With live detection using MTCNN and efficient preprocessing techniques, the proposed algorithm is expected to achieve real-time performance, enabling the rapid and accurate detection of spoofed faces in live video streams.

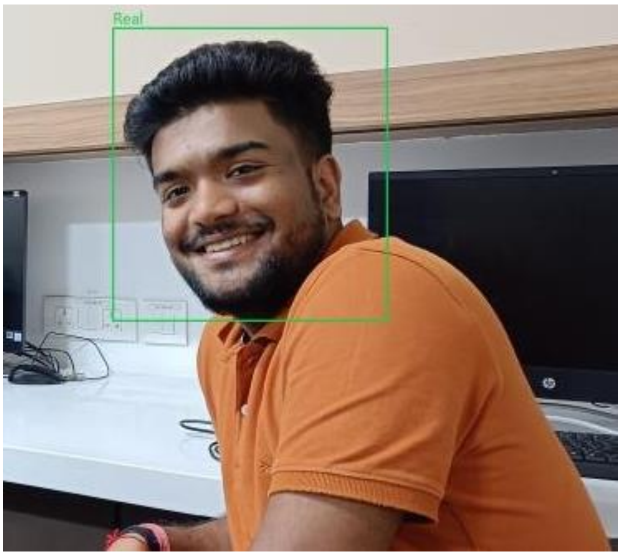


**Figure 7.** Real image output.


**Figure 8.** Spoofed image output.

 

## CONCLUSION

This research has proposed an improved method for face detection for various challenges such as different lighting conditions, various poses, environmental effects, various facial expressions, and backgrounds. A genetic algorithm was used for face detection, and its accuracy was improved. We propose a preprocessing step that includes image enhancement and noise removal. Finally, it can accurately identify the face of a human from the provided image and whether the image is spoofed from the image or through real-time face capture. However, it is not yet suitable for face recognition in video surveillance because of the various drawbacks of surveillance systems. In future work, we intend to find an optimal way to overcome these limitations, and this work will continue in the future for face recognition, matching, and spoofing detection.

## REFERENCES

1. Kumar S, Singh S, Kumar J. A study on face recognition techniques with age and gender classification. 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India. 2017. p. 1001–6. doi:10.1109/CCAA.2017.8229960.
2. Kumar S, Singh S, Kumar J. A comparative study on face spoofing attacks. 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India. 2017. p. 1104–8. doi:10.1109/CCAA.2017.8229961.
3. Kumar S, Deepika, Kumar M. An improved face detection technique for a long distance and near-infrared images. Int J Eng Res Mod Educ. 2017 Jul 10;2(1):176–81. doi: 10.5281/zenodo.825253.
4. Hatem H, Beiji Z, Majeed R. A survey of feature base methods for human face detection. Int J Control Autom. 2015;8:61–78. doi:10.14257/ijca.2015.8.5.07.
5. Ghimire D, Lee J. A robust face detection method based on skin color and edges. J Inf Process Syst. 2013;9:141–56. doi:10.3745/JIPS.2013.9.1.141.
6. Peng S, Ser W, Chen B, Sun L, Lin Z. Robust constrained adaptive filtering under minimum error entropy criterion. IEEE Trans Circuits Syst II. 2018;65:1119–23. doi:10.1109/TCSII.2017.27 89207.
7. Srinivas M, Patnaik LM. Genetic algorithms: A survey. Computer. 1994;27:17–26. doi:10.1109/ 2.294849.
8. Kumar S, Singh S, Kumar J. Automatic face detection using genetic algorithm for various challenges. Int J Sci Res Mod Educ. 2017;2:197–203.
9. Sukhija P, Behal S, Singh P. Face recognition system using genetic algorithm. Procedia Comput Sci. 2016;85:410–7. doi:10.1016/j.procs.2016.05.183.
10. Pratap M, Kumar N. Face recognition using genetic algorithm and neural networks. Int J Comput Appl. 2012;55:975–8887.
11. Musikhin AG, Yu Burenin S. Face recognition using multitasking cascading convolutional networks. IOP Conf Ser Mater Sci Eng. 2021;1155. doi:10.1088/1757-899X/1155/1/012057.
12. Suguna GC, Kavitha HS, Shirahatti S, Bangari SR. Face recognition system for real time applications using SVM combined with FaceNet and MTCNN. Int J Electr Eng Technol. 2021 Jun;12(6):328–35. doi: 10.34218/IJEET.12.6.2021.031.
13. Khan SS, SenGupta D, Ghosh A, Chaudhuri A. MTCNN++: A CNN-based face detection algorithm inspired by MTCNN. Vis Comput. 2024;40:899–917. doi:10.1007/s00371-023-02822-0.
14. Mahmud F, Haque ME, Zuhori ST, Pal B. Human face recognition using PCA based genetic algorithm. 2014 International Conference on Electrical Engineering and Information & Communication Technology, Dhaka, Bangladesh. 2014. p. 1–5. doi:10.1109/ICEEICT.2014. 6919046.
15. Zhi H, Liu S. Face recognition based on genetic algorithm. J Vis Commun Image Represent. 2019;58:495–502. doi:10.1016/j.jvcir.2018.12.012.
16. Chingovska I, Anjos A, Marcel S. On the effectiveness of local binary patterns in face anti-spoofing. 2012 BIOSIG - Proceedings of the International Conference of Biometrics Special Interest Group (BIOSIG); 2012 Sep 6–7; Darmstadt, Germany. p. 1–7.

17. Komulainen J, Hadid A, Pietikäinen M. Context based face anti-spoofing. 2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS), Arlington, VA, USA. 2013. p. 1–8. doi:10.1109/BTAS.2013.6712690.
18. Määttä J, Hadid A, Pietikäinen M. Face spoofing detection from single images using micro-texture analysis. 2011 International Joint Conference on Biometrics (IJCB), Washington, DC, USA. 2011. p. 1–7. doi:10.1109/IJCB.2011.6117510.
19. Galbally J, Marcel S, Fierrez J. Biometric antispoofing methods: A survey in face recognition. IEEE Access. 2014;2:1530–52. doi:10.1109/ACCESS.2014.2381273.
20. Boulkenafet Z, Komulainen J, Hadid A. Face antispoofing using speeded-up robust features and Fisher vector encoding. IEEE Signal Process Lett. 2017;24:141–5. doi:10.1109/LSP.2016. 2630740.
21. Canny J. A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell. 1986;8(6):679–98. doi:10.1109/TPAMI.1986.4767851. PMID: 21869365.
22. Kang WX, Yang QQ, Liang RP. The comparative research on image segmentation algorithms. 2009 First International Workshop on Education Technology and Computer Science, Wuhan, China. 2009. p. 703–7. doi:10.1109/ETCS.2009.417.
23. Verma OP, Parihar AS. An optimal fuzzy system for edge detection in color images using bacterial foraging algorithm. IEEE Trans Fuzzy Syst. 2017;25:114–27. doi:10.1109/TFUZZ.2016.2551289.

 