STM JOURNALS

Current Trends in
Signal Processing

https://journals.stmjournals.com/ctsp

ISSN: 2277-6176 (Online)
ISSN: 2321-4252 (Print)
Volume 15, Issue 1, 2025
January–April
DOI (Journal): 10.37591/CTSP

Review CTSP

Face Detection and Classification for Attendance Systems on Android

Yash Bagul*, Shashikant Chavan, Malhari Shelar, Subodh Raithak

Abstract

This work offers a facial recognition-based attendance system with the goal of addressing the drawbacks of traditional manual attendance. The manual attendance procedure can be made more efficient by using facial recognition technologies and mobile platforms. This design is divided into three function modules: attendance sign-in, attendance record, and face recognition system of check on work attendance information input. It also introduces a face detection and classification principle, analyses the process of building the face recognition classifier, and, finally, designs and implements a face recognition system to check work attendance on the Android platform. The feasibility of this scheme is verified by comparing the experiment results of face recognition accuracy.

Keywords: Android platform, face detection, face recognition, attendance system, mobile platform

INTRODUCTION

The old-fashioned manual attendance system has many drawbacks. It requires human labor, eats up class time, is difficult to enforce rigorously, and is difficult to ensure correctness. There are a lot of defects in the traditional method of manual attendance, for example, it will take up class time, and being operated by people, it is hard to guarantee accuracy, and hard to execute strictly.

We have entered the era of mobile Internet due to the widespread use of the Internet and the quick development of mobile devices. Given that there are more than 5 billion mobile users worldwide, an increasing number of mobile app developers and apps seem to be catering to these users. In this regard, creating an automated attendance system using face recognition technology and a mobile platform might significantly increase productivity. A face's characteristic biological features, which are strong individual differences and self-stability, make it an excellent foundation for identity recognition and authentication. Face features are friendly, dependable, safe, and simple to accept when compared to other approaches. There are a lot of defects in the traditional method of manual attendance, for example, it will take up class time, and being operated by people, it is hard to guarantee the accuracy, and hard to execute strictly [1–6].

***Author for Correspondence**
Yash Bagul
E-mail: yashnbagul@gmail.com

Student, Department of Electronics & Communication Engineering, Parvatibai Genba Moze College of Engineering Wagholi, Pune, India

Received Date: November 25, 2024
Accepted Date: January 07, 2025
Published Date: January 11, 2025

Citation: Yash Bagul, Shashikant Chavan, Malhari Shelar, Subodh Raithak. Face Detection and Classification for Attendance Systems on Android. Current Trends in Signal Processing. 2025; 15(1): 15–22p.

Humans recognize faces so easily that babies as early as three days old can identify familiar faces in their environment. However, facial recognition is a highly challenging task for computers. In the initial automated facial recognition system, the positions of eyes, ears, noses, and other features are marked on a feature vector, which is then used to calculate the Euclidean distance between the feature vectors of images to identify faces. In the research, the geometric properties of facial photos are described for recognition using a 22-dimensional feature vector in a huge database. The study presents the feature face technique, which simplifies the classification problem by treating the facial image as a point that is represented in a low-dimensional space that is derived from a high-dimensional image space.

Classifying species in taxonomy can be challenging because a single characteristic is typically insufficient for accurate identification. To gain a more accurate and comprehensive understanding of an organism, scientists consider various characteristics, including size, shape, color, and genetic information. For instance, an Android-based system for recording classroom attendance may automatically take pictures of students as they enter the room, eliminating the need for manual check-ins or the use of cards. To verify if these faces belong to the class, the system compares them to a pre-registered database of student images. If a match is found, the system immediately marks the student as present. This approach is rapid, secure, and error-free, making it ideal for scenarios where precision and efficiency are crucial [7–11].

DESIGN OF THE SYSTEM

Figure 1 illustrates the three functional modules of the face recognition attendance system: information entry, attendance check-in, and attendance record. The first module is used to input users' basic information while simultaneously gathering user face images. Once the user face image collection is complete, a face recognition classifier is trained on the collected data. The second module performs the system's primary function; if a user is successfully identified by "scanning face", an attendance record is generated; if not, an automated photo for a deposit certificate is taken, and a manual record is made. As seen in Figure 2, the basic face recognition procedure is broken down into three stages: face detection, classifier training, and face recognition. To identify the area of a picture that contains a face is the aim of face detection. The AdaBoost cascade classifier in this procedure first needs to preprocess the input image before it can identify the face portion. The classifier is trained on photos and tags for face identification in the second stage, which involves extracting LBP features from the face region and computing its LBP histogram. To identify each user, third-stage LBP features were retrieved and fed into the classifier based on the observed face region.

Facial Recognition

A crucial component of the face recognition attendance system is face detection. It is employed to determine whether a face is present in a picture. The facial region will be designated for feature extraction and training if this is the case. Face detection in this study is accomplished using the AdaBoost cascade classifier.

Regression and classification tasks are the two main uses for AdaBoost. The approach is iterative in nature, training multiple weak classifiers on a training set before superimposing them to create a strong classifier. The data distribution is changed iteratively to implement the algorithm. Each sample's weight is adjusted based on its classification outcome and the overall accuracy of the last training epoch. The modified data set is then sent to the next classifier for training, and all the classifiers from each training epoch are finally combined to form the final strong classifier [6].

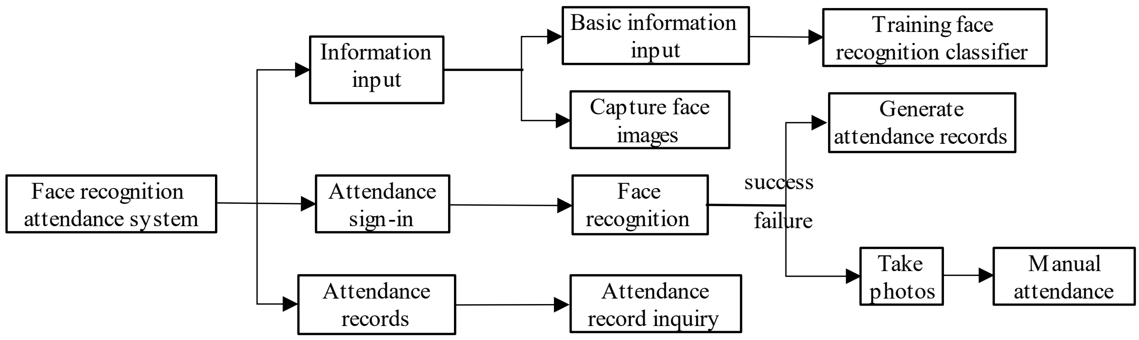

Figure 1. Structure of face recognition attendance system.

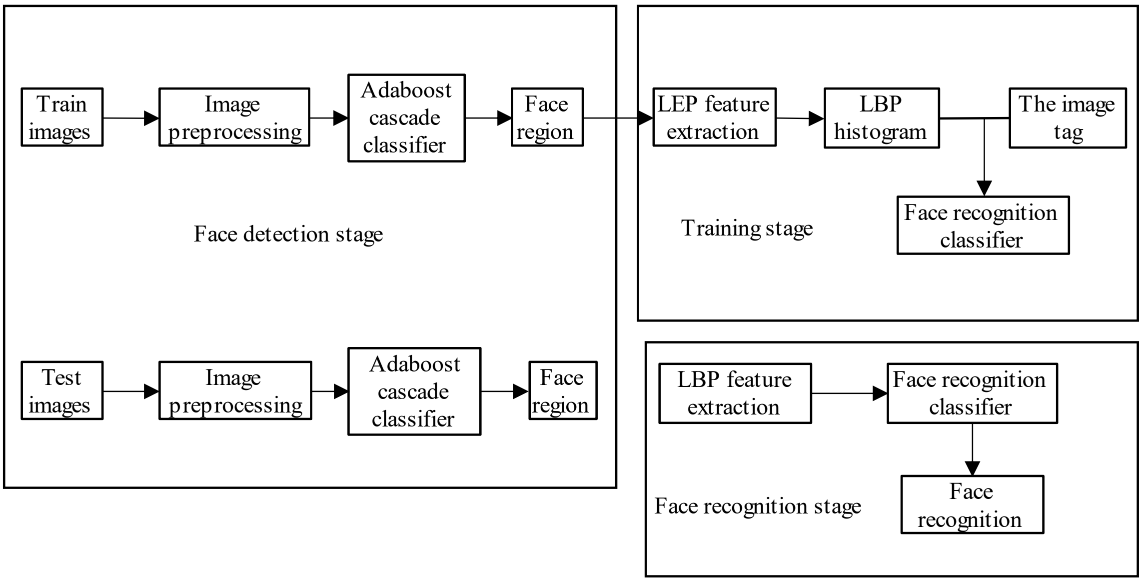

Figure 2. Flow chart of face recognition.

The following illustrates the AdaBoost algorithm's specific steps:

Set the weight of each training sample to 1/N, where N is the total number of samples.

$$D_1 = \{(\omega_{1i} | i = 1, 2, \ldots, N)\}$$

Represent the initial weight distribution, with $\omega_{1i} = \frac{1}{N}$ for all training examples.

When m=1, 2, …, M:

Train a Weak Classifier

Learn weak classifier $G_m(x)$ using the training dataset and the current weight distribution D_m. This classifier $G_m(x)$ is typically a simple model, such as a decision stump (a decision tree with just one level), and the output is either +1 or 1, representing the predicted class.

Calculate the Classification Error

Compute the classification error rate of the weak classifier $G_m(x)$ on the training set, using the current weight distribution D_m:

$$\epsilon_m = \sum_{i=1}^{N} \omega_{mi} \cdot \coprod(G_m(x_i) \neq yi)$$

where $\coprod(\cdot)$ is an indicator function that is 1 if the classifier's prediction is incorrect and 0 otherwise. This error term ϵ_m represents the weighted error rate of the classifier.

Classification Error and Updating Weights

The classification error rate of $G_m(x)$ on the weighted training dataset is the total of the weights of the samples that $G_m(x)$ incorrectly classifies. This error can be computed as:

$$\epsilon_m = \sum_{i=1}^{N} \omega_{m,i} \cdot \coprod(G_m(x_i) \neq yi)$$

After calculating the error, the coefficient α_m for $G_m(x)$ is determined using the formula:

$$\alpha_m = \frac{1}{2} \log\left(\frac{1-\epsilon_m}{\epsilon_m}\right)$$

Update the Weight Distribution

The weights of the training samples are updated to emphasize misclassified samples for the next iteration. The updated weight for each sample *i* is calculated as:

$$\omega_{m+1,i} = \omega_{m,i} \cdot \exp\left(-\alpha_m \cdot yi \cdot G_m(x_i)\right)$$

Where yi is the true label of the sample and $G_m(x_i)$ is the prediction of the weak classifier for that sample. Normalization factor Z ensures that the new weight distribution sums to 1 and is calculated as:

$$Z_m = \sum_{i=1}^{N} \omega_{m,i} \cdot \exp\left(-\alpha_m \cdot yi \cdot G_m(x_i)\right)$$

Final Classifier

After iterating through all M rounds, the final classifier is formed as a weighted sum of all the weak classifiers. The overall prediction is the sign of the weighted combination:

$$H(x) = \sin\left(\sum_{i=1}^{N} \alpha_m \cdot G_m(x)\right)$$

Ultimately, the final classifier is obtained. The weak classifier used is relatively simple, as it doesn't require feature filtering, and issues like overfitting are minimal. However, training can be time-consuming, and the results may vary depending on the choice of the weak classifier. In this study, the AdaBoost cascade classifier, available through OpenCV, was employed. This classifier can be easily unpacked and offers excellent performance, making it an effective choice for this application.

Feature Extraction

Feature extraction is the main problem in face recognition. A classifier can be trained using the extracted facial features, after which the classifier can identify the query image using the same attributes that were extracted.

Extraction of Facial Features

Local Binary Pattern (LBP) was chosen as the face recognition feature in this work [9]. An operator called LBP is used to characterize an image's local texture features. Its rotation invariance, gray invariance, and ease of computation are important benefits.

The fundamental principle of LBP is to take a pixel as the center and perform a threshold comparison with neighboring pixels by adding up the difference between the pixel and its local surrounding pixels. One indicates if the center pixel is brighter than or equal to its neighbor; if not, it is marked as zero. Local binary mode, often known as LBP code, was first implemented in the 3×3 neighborhood, where each pixel was represented by a binary digit, as shown in Figure 3.

The Local Binary Pattern (LBP) algorithm is a method for texture analysis and image feature extraction. It is defined as:

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} s\left(i_p - i_c\right) \cdot 2^p$$

Where:

i_c: The brightness of the central pixel located at (x_c, y_c).

i_p: The brightness of the surrounding pixels.

s(x): The sign function, defined as:

$$s(x) \begin{cases} 1, if\ x \geq 0 \\ 0, if\ x < 0 \end{cases}$$

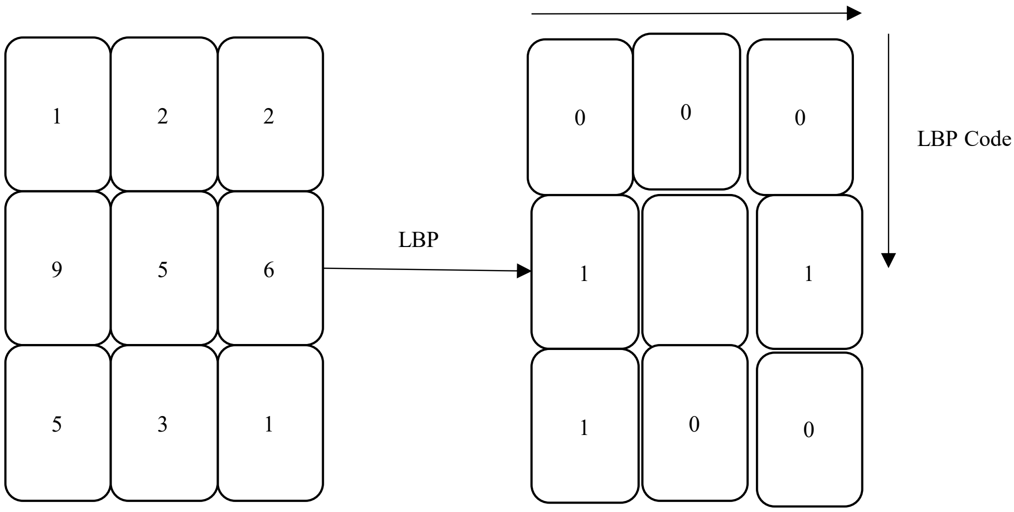

Figure 3. LBP code generation diagram.

This approach effectively captures the local details of an image by comparing the brightness of neighboring pixels to the central pixel. However, it does not account for scale variations as it relies on a fixed surrounding region for coding.

This description approach does a good job of capturing the image's details, but it is unable to leverage the fixed surrounding region for scale change coding. A variable extension approach is proposed in the literature that uses a circle with a variable radius to encode the pixels closest to the neighbor such that the nearest neighbor, depicted in Figure 4, can be recorded:

The following formula can be used to find pęP for a given location (xc, yc), its nearest neighbor point.

$$x_p = x_c + R \cdot \cos(p)\,, y_p = y_c - R \cdot \sin(p)$$

Where,

R: Radius of the circle.

p: Angular position of the neighboring pixel in the circle.

(x_p, y_p): The coordinates of the neighboring pixel.

This operation is referred to as Extended LBP or Circular LBP, which enhances the standard LBP operator by encoding scale changes.

This operation, sometimes referred to as extended LBP or circular LBP, is an expansion of the initial LBP operator. Interpolation points are utilized for circle points that are not on the image coordinates. The bilinear interpolation from the OpenCV library was used in this study, and it was described as follows:

$$f(x,y) = f(0,0)(1-x)(1-y) + f(0,1)(x)(1-y) + f(1,0)(1-x)(y) + f(1,1)(x)(y)$$

Where, f (x, y) is the interpolated value at the coordinates (x, y).

Classifier for Face Recognition

The representation approach described in the literature is used in this work, and the USES statistical histogram of the LBP feature spectrum is used as the feature vector for recognition and classification. The first phase involves segmenting an image into many regions and extracting LBP features for each pixel inside a region. Following that, each region's statistical histogram of LBP features. A statistical histogram can thus be used to characterize a region, and the entire image is made up of many histograms, also referred to as local binary mode histograms.

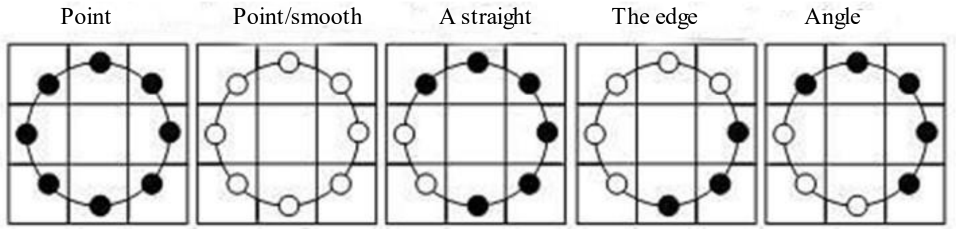

Figure 4. Schematic diagram of extended LBP nearest neighbor coding.

To some extent, the errors resulting from inadequate image alignment can be minimized by segmenting the image into sections and computing the LBP histogram individually. In the interim, certain locations may be given distinct weights. For instance, the center portion can be given a higher weight than the edge component, meaning that when it comes to image matching and identification, the central part matters more. To build a face recognition classifier, photos must be trained and tested. The facial image and its associated label should be input simultaneously during training. In the face detection stage, the LBP feature of a face region was extracted, and LBP histograms were created. The face recognition classifier then uniformly extracts the LBP features of these images, statistics the LBP histograms, and takes the feature vectors composed of these histograms as the feature vector of this face. This histogram's distance from each of the classifier's other feature vectors is determined by the similarity measure function; in the interim, an experimental threshold value is determined.

Ultimately, the test image's classification result is determined by selecting the face image that has the least distance and the lowest threshold value. In this study, two facial photos are compared using the following similarity calculation formula:

$$d(H_1, H_2) = (H_1 \cdot I_1 - H_2 \cdot I_2)^2 \cdot (H_1 \cdot I_1)$$

RESULTS OF EXPERIMENTS

The project made use of Android devices. You can choose from the mobile album or take pictures with the camera for training and testing. The tests in this research were performed using the face database maintained by the Olivetti Research Laboratory. This collection, which includes 400 grayscale photos of people of various ages, genders, and ethnicities, was produced by Olivetti Laboratory in Cambridge, UK. Each one comprises 10 pictures, each featuring a different set of details and facial emotions, like smiling or frowning, having eyes open or closed, wearing or not wearing spectacles, etc. Figure 5 displays a selection of ORL face database example photographs.

Two traditional facial recognition algorithms, Fisher face and Eigenface are tested concurrently to confirm the correctness of the LBP algorithm [8]. The initial set of N images *(N=1, 2...9)* served as training samples, whereas the final set of images served as test samples. Figure 6 presents a comparison of the three algorithms' experimental findings. Furthermore, the time spent training a face recognition classifier using three distinct methods under varying picture counts is also measured in this work. The statistical findings are displayed in Table 1.

Figure 6 illustrates how face recognition performance increased dramatically as the number of training images increased. Face recognition can achieve over 95% accuracy when evaluated on five different photos per subject. The LBP method performed better than the other two algorithms when there were three or more photographs per person. Table 1 shows that the LBP approach outperforms the other two techniques in terms of time efficiency throughout the face recognition classifier's training phase. We conducted an experiment on 30 people to confirm the face recognition attendance system's viability. For the "information input" module, we collected 10 photos for every trainee and used face recognition for "attendance check-in" after that. Ultimately, it is possible to accurately identify every individual, confirming the system's utility [10].

Figure 5. ORL face database partial instance image.

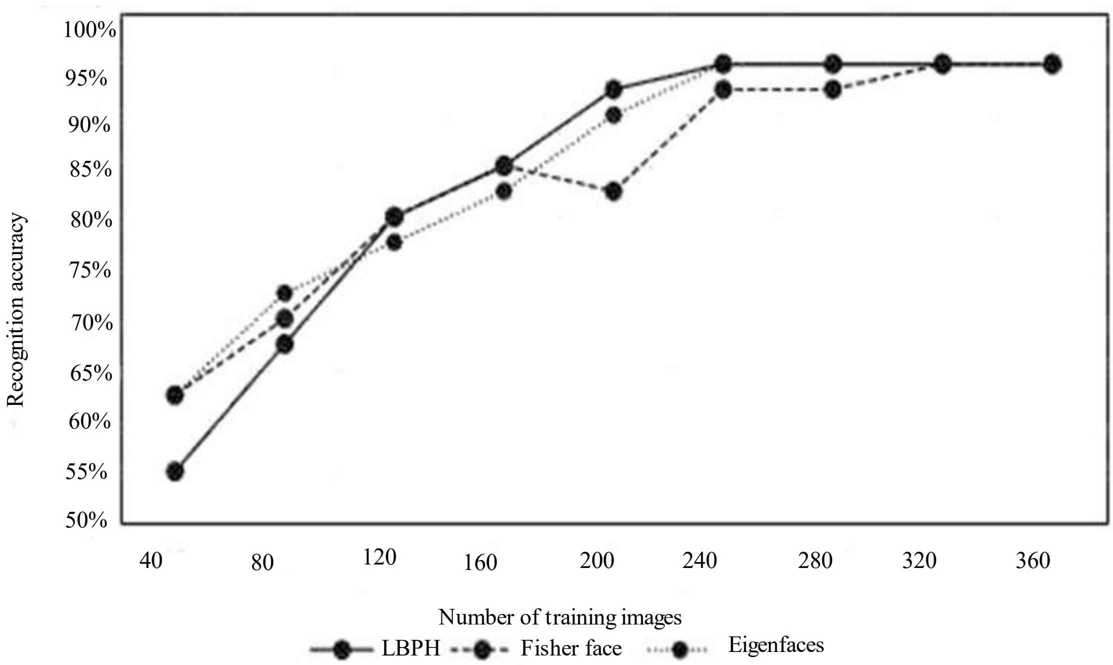

Figure 6. Statistical face recognition accuracy of the three algorithms in ORL library.

Table 1. Statistical training time (ms) of three algorithms for different numbers of images.

Number of training images	40	120	200	280	360
LBP	759	1681	2681	3818	4759
Fisher face	2212	7994	16992	37052	72292
Eigenface	2161	6612	21638	31025	55918

CONCLUSION

The AdaBoost cascade classifier is used in this ’study’s face detection process on the Android platform to identify the face portion. In the training stage, a face image is extracted using LBP features, a histogram is computed, and a face recognition classifier is built using the LBP histogram. In the recognition stage, additional LBP features are extracted and fed into the classifier. Finally, a similar measurement function is computed to efficiently accomplish personnel identification. This study presents the design and implementation of a face recognition attendance system that is evaluated on the ORL face database and compared with the Fisher face and Eigenfaces algorithms. Finally, a final test with 30 testers confirms that the system’s facial recognition accuracy can satisfy the practical usage requirement.

REFERENCES

1. Sunaryono D, Siswantoro J, Anggoro R. An android based course attendance system using face recognition. J King Saud Univ-Comput Inf Sci. 2021 Mar 1; 33(3): 304–12.
2. Bai X, Jiang F, Shi T, Wu Y. Design of attendance system based on face recognition and android platform. In 2020 IEEE International Conference on Computer Network, Electronic and Automation (ICCNEA). 2020 Sep 25; 117–121.
3. Nugraha AA, Kuswardhana D. Online Class Attendances System for Remote Learning Monitoring Using Face Recognition Technology Based on Mobile App with Android Studio Platform. 2021 Sep.
4. Samet R, Tanriverdi M. Face recognition-based mobile automatic classroom attendance management system. 2017 International Conference on Cyberworlds (CW), Chester, UK. 2017. pp. 253-256. DOI: 10.1109/CW.2017.34.
5. Alhayani BS, Rane P. Face recognition system by image processing. Int J Electron Commun Eng Technol. 2014 May 30; 5(5): 80–90.
6. Brunelli R, Poggio T. Face recognition through geometrical features. In Computer Vision—ECCV'92: Second European Conference on Computer Vision Santa Margherita Ligure, Italy; May 19–22, 1992 Proceedings 2. Berlin Heidelberg: Springer; 1992; 792–800.
7. Turk M, Pentland A. Eigenfaces for recognition. J Cogn Neurosci. 1991 Jan 1; 3(1): 71–86.
8. Belhumeur PN, Hespanha JP, Kriegman DJ. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Trans Pattern Anal Mach Intell. 1997 Jul; 19(7): 711–20.
9. Rahim MA, Hossain MN, Wahid T, Azam MS. Face recognition using local binary patterns (LBP). Glob J Comput Sci Technol. 2013 May; 13(4): 1–8.
10. Ojala T, Pietikainen M, Maenpaa T. Multiresolution grayscale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell. 2002 Jul; 24(7): 971–87.
11. Fisher RA. The use of multiple measurements in taxonomic problems. Ann Eugen. 1936 Sep; 7(2): 179–88.