Journal of Physics: Conference Series

Spoofing detection on facial images recognition using LBP and GLCM combination

ECS Toyota Young Investigator Fellowship
ECS
TOYOTA
For young professionals and scholars pursuing research in batteries, fuel cells and hydrogen, and future sustainable technologies.
At least one $50,000 fellowship is available annually.
More than $1.4 million awarded since 2015!
Application deadline: January 31, 2023
Learn more. Apply today!

Spoofing detection on facial images recognition using LBP and GLCM combination

F Sthevanie, K N Ramadhani
Multimedia Laboratory, School of Computing, Universitas Telkom, Jl. Telekomunikasi No. 1, Terusan Buah Batu Bandung 40257, Indonesia

sthevanie@telkomuniversity.ac.id, kurniawannr@telkomuniversity.ac.id

Abstract. The challenge for the facial based security system is how to detect facial image falsification such as facial image spoofing. Spoofing occurs when someone try to pretend as a registered user to obtain illegal access and gain advantage from the protected system. This research implements facial image spoofing detection method by analyzing image texture. The proposed method for texture analysis combines the Local Binary Pattern (LBP) and Gray Level Co-occurrence Matrix (GLCM) method. The experimental results show that spoofing detection using LBP and GLCM combination achieves high detection rate compared to that of using only LBP feature or GLCM feature.

1. Introduction

Nowadays, the face detection system has been widely implemented in daily life. The face detection system usually used in security system because of the unique features on every human faces. However, the face detection system has some drawbacks such as image spoofing. Facial image spoofing is an attack that deceive the face detection system using imprinted images. Spoofing attack occurs when someone try to impersonate a registered user by forging the face image and taking advantage of such illegal access[1]. The biometric system becomes vulnerable without the spoofing detection mechanism.

Original facial images and imprinted facial images reflect light in different ways. Original face is a complex 3D-object where the imprinted face image is rigid planar object. Other differences is that human face has pigment while imprinted face image does not have any pigment. Using those differences, we can build a system that can detect spoofing attack on facial biometric system.

Texture analysis is one of the fundamental research in computer vision and is an important component in the pattern recognition and classification. Texture analysis is commonly used for image classification, content-based image retrieval, 3D image reconstruction and image segmentation. There were a lot of research used texture analysis to detect spoofing attack in facial biometric system. The method used were Local Binary Pattern (LBP), Gabor Wavelet, Haboor Wavelet, Histogram of Oriented Gradient (HOG), Spatiotemporal Local Binary Pattern[2,3,4,5,6]. The ability of LBP to extract image features was used to distinguish the original human face images and the imprinted face images. Another feature extraction used to detect spoofing was GLCM which able to extract the contrast level of an image. The contrast level was used to distinguish spoof or non-spoof images. This study compares the performance of Local Binary Pattern and Gray-Level Co-Occurrence Matrix methods as a feature extraction method to classify the image into two classes i.e. spoof and non-spoof classes. We combine the LBP and GLCM method to extract the features used to detect the spoofing images. The LBP and GLCM are based on texture feature, but GLCM uses statistical function so the extraction result represents the overall image texture features. By combining the

LBP histogram and GLCM statistical function, we expect to obtain a better result compared to by using only one feature function.

2. Related works

The attacks on biometric can be divided into several scenarios. The attack can be a forcing to a registered users, registering to the system using deceased person, using truncated body parts with a genetic clone, or by spoofing the data of registered user. The biometric characteristics of each individual registered user must be different even for identical twins. Using only algorithm is not strong enough to counter those differences[7]. This research is focused on spoofing attack.

Spoofing occurs when someone wants to fool the system by faking the existing data which in this case is the face image of registered user. The attacker uses an imprinted facial image of registered user to enter the biometric security access and then gain access of the system. Without the anti-spoofing mechanism, the recognition system is very vulnerable to be attacked. Anti-spoofing mechanism can be built using liveness detection. The spoofing detection can be built using some liveness sign to differentiate between the original face and the imprinted image. Human face has a complex 3D features that are not rigid, while the imprinted face image is a rigid planar object. We can use texture features to extract those differences. Thus, we can use texture analysis to detect spoofing attacks on human face images[8].

3. Local Binary Pattern (LBP)

Local Binary Pattern (LBP) method was proposed to describe the texture of an image. LBP was used in many researchs in computer vision such as face recognition, facial experession recognition, motion and action modeling and in medical image analysis. The LBP is derived from the neighboring radius of a given pixel. The LBP value is calculated by comparing the intensity of neighboring pixel to the intensity of the central pixel starting from a certain angle and repeat it on the next neighboring pixel[9]. The comparison result is a binary number of 1 and 0. A value 1 is given if the center pixel intensity is less than the intensity of the neighboring pixel, whereas a value of 0 is given if the center pixel intensity is more than the intensity of the neighboring pixel. The most commonly used sample is P = 8 with the circle radius R = 1. P = 8 means that we conduct the LBP on 8 neighboring pixel around the center pixel. R = 1 means that the distance between the center pixel and neighboring pixel is 1. However, other number of neighboring pixel and radius values such as P = 16 and R = 2 can be used. After that, we concate the binary value of the neighboring pixel clockwise or counterclockwise. The binary value is then converted to decimal value. This process is conducted for every pixel in the image. The illustration of LBP process can be seen on Figures 1 and 2.

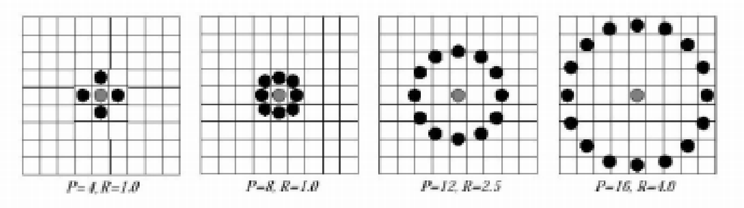


Figure 1 LBP Illustration for different P and R values

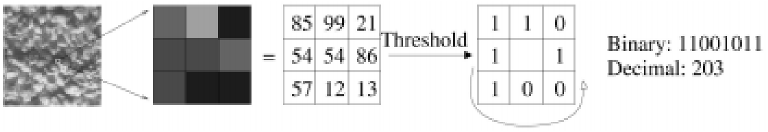


Figure 2 LBP Encoding Example

The LBP code of a pixel (x_c, y_c) [10] can be calculated by:

$$LBP_{P,R} = \sum_{P=0}^{P-1} s(g_p - g_c)2^p, \tag{1}$$

where g_c is the intensity value of the central pixel (x_c, y_c), g_p is the intensity value of the neighboring pixel P in radius R, and s is defined as the thresholding function as:

$$s(x) = \begin{cases} 1, if\ x \geq 0; \\ 0, otherwise. \end{cases} \tag{2}$$

The feature extraction using LBP is divided into two stages.

1. Build LBP matrix by generating every LBP value for every pixel. The example of LBP matrix can be seen on Figure 3.

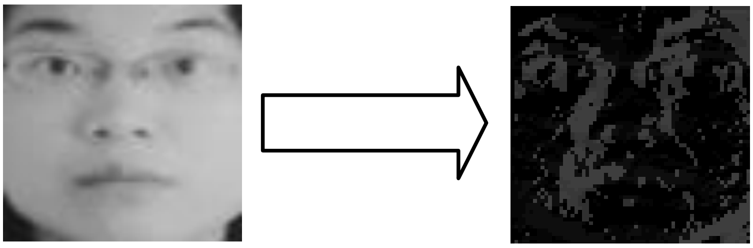


Figure 3 The Example of LBP Matrix (right) from original image (left)

2. Calculate histogram from LBP matrix. On this stage, we build histogram that contain the number of pixel for every LBP intensity level. The number of LBP intensity level is depend on the number of neighboring pixel (P value), that is 2^P. On this research, we use P=8, so the number of LBP intensity level is 256. The example for LBP histogram can be seen on Figure 4.

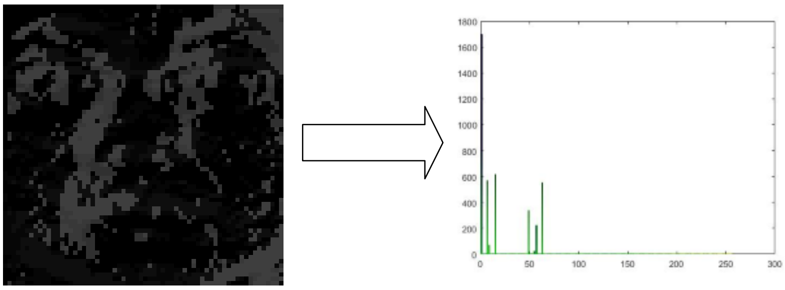
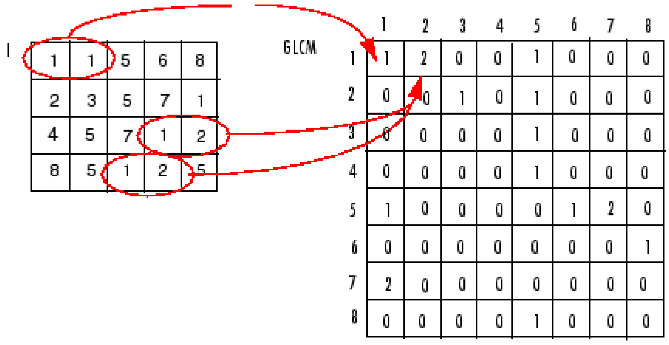


Figure 4 Example of LBP Histogram (right) from LBP Matrix (left)

4. Gray-Level Co-Occurrence Matrix (GLCM)

The texture analysis approach has been widely developed in recent decades, and can be classified into four categories such as statistical, geometrical, model-based and signal processing. Gray Level Co-occurrence Matrix (GLCM) has been shown to be one of the most efficient approaches to texture analysis among other statistical methods[11]. GLCM describes the spatial distribution of the gray value of an image and the frequency of gray occurrences pattern with a specified angles and distance. GLCM can be used for recognition and texture classification. Texture analysis using Gray Level Co-occurence Matrix (GLCM) was introduced by Haralick[12]. GLCM is a matrix sized NxN where N is the grayscale intensity level. Each of matrix element value located on (i,j) is calculated from the number pattern from original grayscale image that contain a specified pattern. The specified pattern contains the gray pixel i neighboring with gray paxel j for distance d and angle θ. The parameter d represents the distance of two neigboring pixel in the grayscale image and θ is the size of the discrete angle (0^0, 45^0, $90\ ^0$, 135^0). The commonly angle value used in GLCM calculations is θ = 45^0 or the mean of the four corners (θ = 0^0, 35^0, 90^0, 135^0). The illustration of GLCM can be seen on Figure 5.



1	1	5	6	8
2	3	5	7	1
4	5	7	1	2
8	5	1	2	5

	1	2	3	4	5	6	7	8
1	1	2	0	0	1	0	0	0
2	0	0	1	0	1	0	0	0
3	0	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0	0
5	1	0	0	0	0	1	2	0
6	0	0	0	0	0	0	0	1
7	2	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0

Figure 5 The Illustration of GLCM method

Based on the GLCM definition, Haralick defines several functions to describe the textual features from GLCM, such as[11]:

1. *Angular Second Moment* (ASM),
$$f_1 = \sum_i \sum_j \{p(i,j)\}^2 \tag{3}$$
2. *Contrast*,
$$f_2 = \sum_{n=0}^{N_g-1} n^2 \left\{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j)\right\}, |i-j| = n \tag{4}$$
3. *Correlation*,
$$f_3 = \frac{\sum_i \sum_j (ij)p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y} \tag{5}$$
4. *Variance*,
$$f_4 = \sum_i \sum_j (i - \mu)^2 p(i,j) \tag{6}$$
5. *Inverse Difference Moment*,
$$f_5 = \sum_i \sum_j \frac{1}{1+(i-j)^2} p(i,j) \tag{7}$$
6. *Sum Average*,
$$f_6 = \sum_{i=2}^{2N_g} i p_{x+y}(i) \tag{8}$$
7. *Sum Variance*,
$$f_7 = \sum_{i=2}^{2N_g} (i - f_8)^2 p_{x+y}(i) \tag{9}$$
8. *Sum Entropy*,
$$f_8 = -\sum_{i=2}^{2N_g} p_{x+y}(i) \log\{p_{x+y}(i)\} \tag{10}$$
9. *Entropy*,
$$f_9 = -\sum_i \sum_j p(i,j) \log(p(i,j)) \tag{11}$$
10. *Difference Variance*,
$$f_{10} = variance\ of\ p_{x-y} \tag{12}$$
11. *Difference Entropy*
$$f_{11} = -\sum_{i=0}^{N_g-1} p_{x-y}(i) \log\{p_{x-y}(i)\}. \tag{13}$$

$p(i,j)$ is the normalized value of GLCM element located on (i,j), $p(i,j) = P(i,j)/R$. $\mu_x, \mu_y, \sigma_x, \sigma_y$ are the mean values and variance values for P_x dan P_y. μ is the mean value for image P. $p_x(i)$ is the value of marginal-probability matrix calculated by summing the row value from $p(i,j), = \sum_{i=1}^{N_g} P(i,j)$. N_g is the *gray levels* used.

$$P_y(j) = \sum_{i=1}^{N_g} p(i,j) \tag{14}$$

$$P_{x+y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j),\ i+j = k;\ k = 2,3,\dots,2N_g \tag{15}$$

$$P_{x-y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j),\ |i-j] = k;\ k = 0,1,\dots,N_g - 1 \tag{16}$$

5. K-Nearest Neighbour

K-Nearest Neighbors (K-NN) is one of the simplest algorithms and has long been used in classification. In some cases, K-NN method provides better results compared to other classifier. The K-NN method classifies the unlabeled vector (testing data) by assigning the most-appearing labels between the nearest training distance k around the testing data. Therefore, the performance of this method is highly dependent on the distance matrix used to identify its nearest neighbor[3]. The distance function usually used in K-NN are euclidean distance, minkowski distance, mahalanobis distance, city block distance, hamming distance, and mahalanobis distance. The distance function commonly used for continuous variables is the euclidean distance. For discrete variables, such as text-based data, other distance calculation methods can be used, such as Hamming Distance. The distance calculation using the euclidean distance is obtained by comparing each value in the training data vector with each value in the testing data vector as can be seen in:

$$d(a,b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \tag{17}$$

6. Research methodology

This research consist of two steps. The first step is modelling. The modelling step is conducted to build the model for texture feature of the training data (spoof and non-spoof) and insert the features to the database. The output of this step is the database of texture feature of training data. The flowchart of modelling step can be seen on Figure 6.

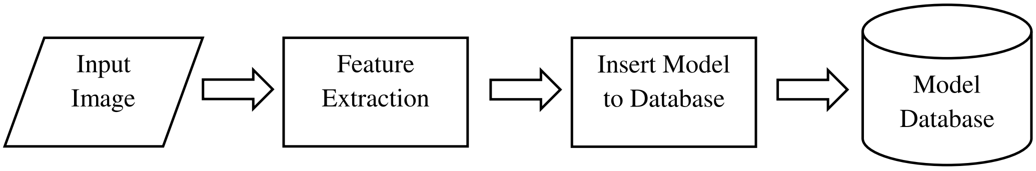


Figure 6 Modelling Step

The second step is testing step. The testing step is an identification process to classify the testing data to spoof or non-spoof label. The step start with input image of testing data. The feature extraction process is conducted on training data to obtain the feature vector. Then, the classification process using K-NN algorithm is conducted. The classification process uses the model database built from modelling step. The output of the classification process is the conclusion whether the data is labelled spoof or non-spoof. The process can be seen on Figure 7.

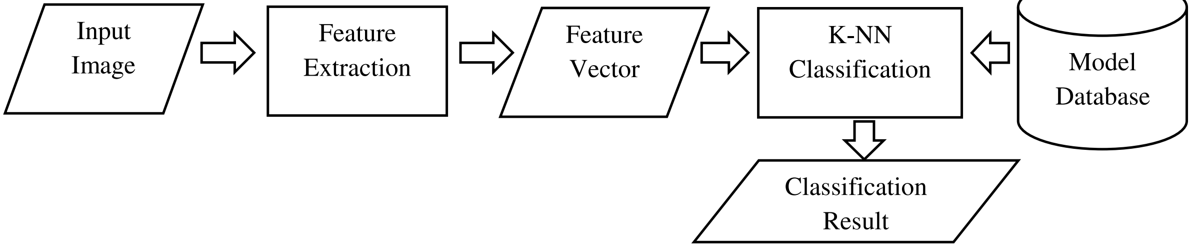


Figure 7 Testing Step

The dataset used in this research are from NUAA dataset. The NUAA database images consist of 15 user images and the number of overall images are 12.614 images of real face and impostor face images. The images were resized to 64 x 64 pixels. The color representation of the images are grayscale with 256 levels.

7. Testing result

There are three scenario for testing process. The first scenario is spoofing detection using LBP algorithm. The second scenario is spoofing detection using GLCM algorithm. The final scenario is spoofing detection by combining LBP and GLCM algorithm.

7.1 Spoofing Detection using LBP Algorithm.

On this step, we conduct testing on spoofing detection using LBP algorithm to determine the best R value from 1 to 5 using value $P = 8$. The result can be seen on Figure 8.

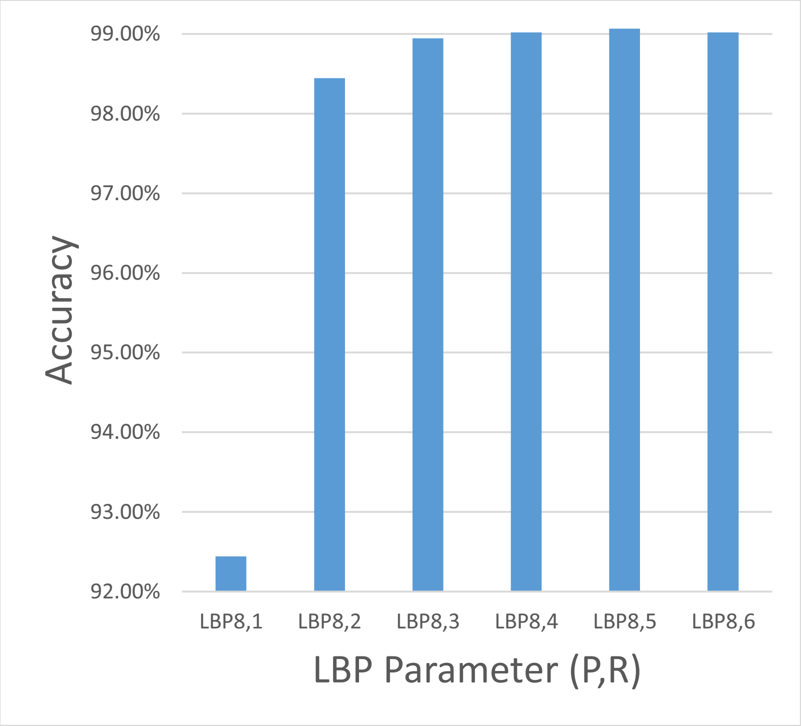


Figure 8 LBP Testing Result

The x-axis shows the different value for LBP radius (R). The y-axis shows the accuracy of spoofing detection using given LBP radius. From Figure 8, can be seen that the overall performance of LBP parameter configuration were more than 90%. The best accuracy was obtained by using R=5 where the accuracy was 99.07%.

7.2 Spoofing Detection using GLCM Algorithm.

On this step, we conduct testing on spoofing detection using GLCM algorithm. We use two mechanism on GLCM algorithm.

1) GLCM1 is implemented by using GLCM on every part of human face image.
2) GLCM2 is implemented by using GLCM only on eyes and nose area. We use only the area that has a contour difference on face image.

The result can be seen on fig 9.

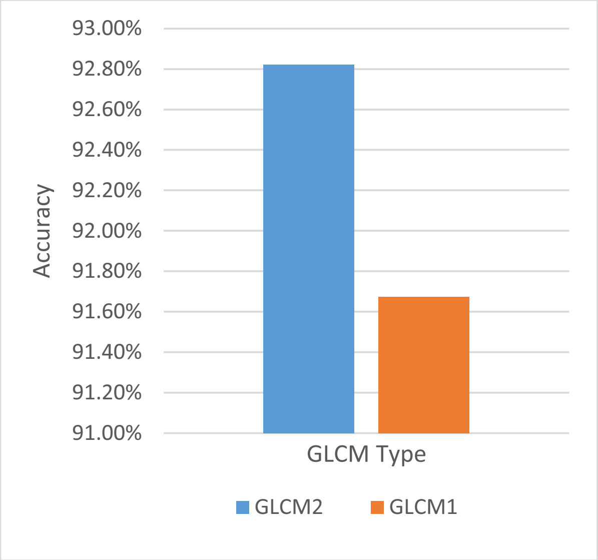


Figure 9 GLCM Testing Result

From Figure 9, we can see that the accuracy of the overall performance spoofing detection using GLCM for the two configuration obtained the accuracy more than 90%, but the performance of GLCM still less than the performance of LBP algorithm. The best accuracy was 92.82% using GLCM2 configuration (using GLCM only in eyes and nose area).

7.3 Spoofing Detection using LBP and GLCM Combination

We combine the LBP and GLCM feature using several scenario.

1) Combination 1. We combine the result of K-NN algorithm on LBP features and GLCM features.
2) Combination 2. We conduct GLCM algorithm on LBP matrix.
3) Combination 3. Similar to combination 2, but the LBP matrix is constructed only from eyes and nose area.
4) Combination 4. Similar to combination 1, but the LBP matrix and GLCM matrix constructed only from eyes and nose area.

The best accuracy from four combination was obtained using combination 4. The best accuracy was 99.26% obtained by combining the K-NN classification result on LBP and GLCM features. The detail of testing result can be seen on Figure 10.

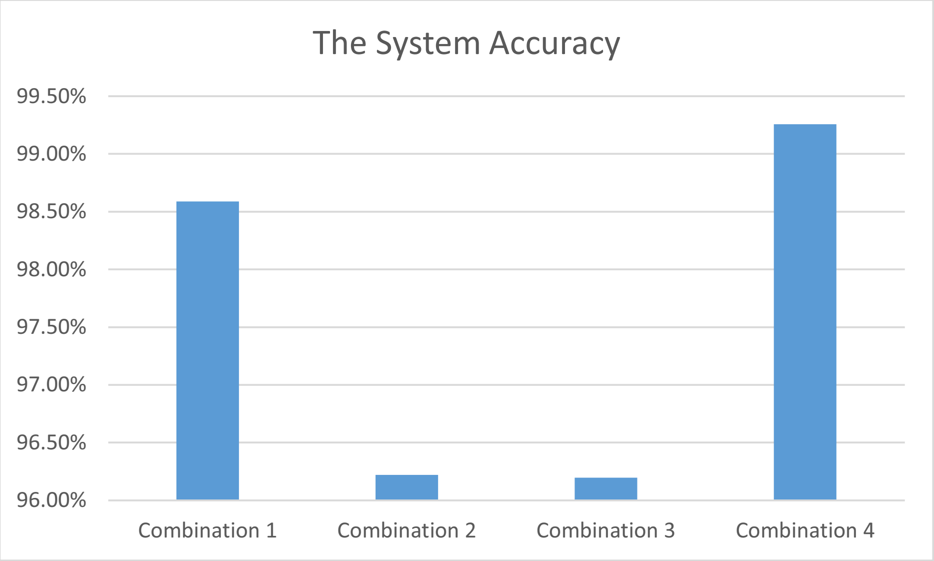


Figure 10 Testing Result on LBP and GLCM Combination

Furthermore, we compared the result of the best combination scenario and the result of LBP algorithm. We compared the error rate for the two scenario. The result can be seen on Figure 11.

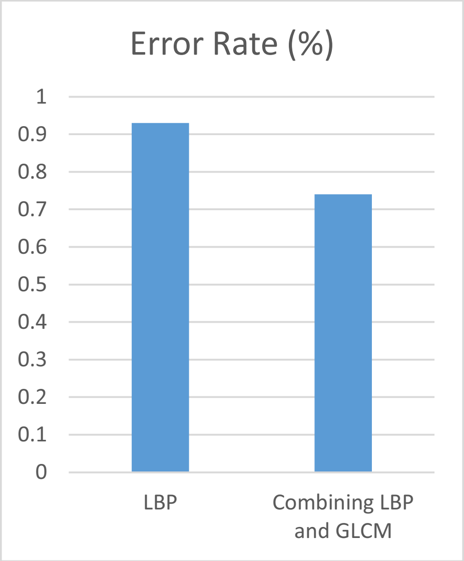


Figure 11 Comparing Error Rate Between LBP and Combining LBP+GLCM

The error rate of spoofing detection using LBP and GLCM combination feature extraction is less than that of feature extraction using only LBP. By using GLCM, we reduced the error rate on LBP feature from 0.93% to 0.74% or the error rate reduced to 20.4%. Thereby, we conclude that the spoofing detection using combination of LBP and GLCM feature extraction is better than using only one feature extraction method, such as LBP and GLCM.

8. Conclusion

In this research, we implemented the spoofing detection on facial biometric using texture features. We implemented two texture feature extraction algorithm, those were LBP and GLCM. The research showed that the spoofing detection using LBP algorithm has a better accuracy than that of using GLCM algorithm.

The LBP algorithm gave the accuracy 99.07% while the GLCM algorithm only gave accuracy 92.82%. The research also showed that the eyes and nose area has an important feature to detect the spoofing on facial biometric. The accuracy of using only eyes and nose area was 92.82% while the accuracy of using every part on human face was 91.67%. By using the combination of LBP and GLCM algorithm for feature extraction algorithm, the spoofing detection system accuracy was 99.26%. That performance was better than that of using only a single feature extraction algorithm, such as LBP and GLCM.

References

[1] Maatta, J., Hadid, A., & Pietikainen, M. (2012). Face spoofing detection from single images using texture and local shape analysis. *Biometrics, IET*, *1*(1), 3-10.

[2] X.Tan, Y.Li, J.Liu and L.Jiang. Face Liveness Detection from A Single Image with Sparse Low Rank Bilinear Discriminative Model, In: Proceedings of 11th European Conference on Computer Vision (ECCV'10), Crete, Greece. September 2010

[3] Komulainen, J., Hadid, A., & Pietikäinen, M. (2012, November). Face spoofing detection using dynamic texture. In *Computer Vision-ACCV 2012 Workshops*(pp. 146-157). Springer Berlin Heidelberg.

[4] Määttä, J., Hadid, A., & Pietikainen, M. (2011, October). Face spoofing detection from single images using micro-texture analysis. In *Biometrics (IJCB), 2011 international joint conference on* (pp. 1-7). IEEE.

[5] Wen, D., Han, H., & Jain, A. K. (2015). Face spoof detection with image distortion analysis. *Information Forensics and Security, IEEE Transactions on*,*10*(4), 746-761.

[6] de Freitas Pereira, T., Anjos, A., De Martino, J. M., & Marcel, S. (2012, November). LBP− TOP based countermeasure against face spoofing attacks. In *Computer Vision-ACCV 2012 Workshops* (pp. 121-132). Springer Berlin Heidelberg.

[7] Schuckers, S. A. (2002). Spoofing and anti-spoofing measures. *Information Security technical report*, *7*(4), 56-62.

[8] Mansfield, A. J., & Wayman, J. L. (2002). *Best practices in testing and reporting performance of biometric devices* (pp. 1-36). Teddington, Middlesex, UK: Centre for Mathematics and Scientific Computing, National Physical Laboratory.

[9] Lahdenoja, O., Poikonen, J., & Laiho, M. (2013). Towards understanding the formation of uniform local binary patterns. *ISRN Machine Vision*, *2013*.

[10] Hadid, A. (2008, November). The local binary pattern approach and its applications to face analysis. In *Image Processing Theory, Tools and Applications, 2008. IPTA 2008. First Workshops on* (pp. 1-9). IEEE.

[11] Haralick, R. M., Shanmugam, K., & Dinstein, I. H. (1973). Textural features for image classification. Systems, Man and Cybernetics, IEEE Transactions on, (6), 610-621.

[12] Hu, Y., Zhao, C. X., & Wang, H. N. (2008, December). Directional analysis of texture images using gray level co-occurrence matrix. In Computational Intelligence and Industrial Application, 2008. PACIIA'08. Pacific-Asia Workshop on (Vol. 2, pp. 277-281). IEEE.

[13] Jakkula, V. (2006). Tutorial on support vector machine (svm). School of EECS, Washington State University.

[14] Cristianini, N., & Shawe-Taylor, J. (2000). An introduction to support vector machines and other kernel-based learning methods. Cambridge university press.