electronics

MDPI

Article

An Enhanced LBPH Approach for Face Recognition Affected by Ambient Light in Data Transmission Planning

Yeong-Chin Chen [1,*], Yi-Sheng Liao [1], Hui-Yu Shen [2], and Yue-Jun Shen [1]

[1] Department of Computer Science and Information Engineering, Asia University, Taichung 413, Taiwan
[2] Department of Information and Network Communications, Chienkuo Technology University, Changhua 500, Taiwan
* Correspondence: ycchenster@gmail.com, neuro168@gmail.com

Abstract: Changing the ambient lighting will blur the contours of objects in the image, which greatly impacts subsequent identification especially in many outdoor face recognition applications (such as face recognition payment). Therefore, it is important to find an algorithm that can capture the major features of the object with ambient light changes. In this paper, image recognition takes face recognition as an example to analyze the differences between Local Binary Patterns Histograms (LBPH) and OpenFace deep learning neural network algorithms and compares the accurate and error rates of face recognition in different environmental lighting. According to the prediction results of 13 images based on grouping statistics, the accurate rate of face recognition of LBPH is higher than that of OpenFace in scenes with changes in ambient lighting. When the azimuth angle of the light source is more than +/- 25° and the elevation angle is +000°, the accurate rate of face recognition is low. When the azimuth angle is between +25° and -25° and the elevation angle is +000°, the accurate rate of face recognition is higher. Through the experimental design, the results show that in concern with the uncertainty of illumination angles of lighting source, the LBPH algorithm does have the advantage of higher accuracy in face recognition. In terms of the ability to recognize and deliver informative data images, the data picture produced has become a feasible solution to the problem of various CCTVs data transmission.

Keywords: LBPH; OpenFace; ambient light; face recognition

Citation: Lastname, F.; Lastname, F.; Lastname, F. Title. *Electronics* **2022**, *11*, x. https://doi.org/10.3390/xxxxx

Academic Editor: Firstname Lastname

Received: date
Accepted: date
Published: date

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

In industrial manufacturing, manufacturers produce large numbers of products by orders, but the manufacturing process will cause partial defects in the products. In the past with underdeveloped technology, manual inspection for product defects was always needed. With the development of imaging technology in recent years, it is possible to automatically screen products for defects through Automatic Optical Inspection (AOI). AOI will classify the flaws into different types, trace the cause of the defects generated by the production machine, and adjust the machine's parameters to reduce the incidence of defects [1]. There are several common steps in image recognition: image capture, image preprocessing, model training, image prediction, and output results of recognition [2]. However, in the process of AOI screening for defects, a lack of light sources around the product may result in unclear lines or features of the defect when extracting the product's image, affecting subsequent identification. Therefore, in the process of AOI, additional light sources kept within a certain range to enhance the lighting of the product is crucial.

Face recognition has become a major interest in AOI image processing and computer vision because of its non-invasiveness and easy access. Extracting face features with good discrimination and robustness, and constructing efficient and reliable classifiers has always been the focus of face recognition research [3].

With the breakthrough of Artificial Intelligence (AI) algorithms, the convenient data collection by the development of the Internet, and the continuous improvement of hardware performance, the accuracy of the face recognition model is greatly improved. The accuracy of face recognition by AI even surpasses that of humans. In recent years, face recognition applications have sprung up prosperously in Taiwan and all the world such as: M-Police face recognition in Police Department, smartphone's Face ID face unlocking [4], Entry and exit systems of unmanned shops, library book checkout systems, airport entry and exit systems, etc. [5] [6] .

Most commercially available automatic systems currently use fixed Closed-Circuit Television (CCTV) cameras, which enables for the deployment of efficient identification and tracking algorithms, as shown in Figure 1. Following the acquisition of meaningful data by CCTV camera, it is critical to ensure that the control room receives the same and authentic data in order to take any action or send an alarm signal to various departments. Unfortunately, even high-quality CCTV cameras have a resolution of 720x480, with low-end cameras having a far lower resolution. This makes activities requiring high resolution and picture quality, such as face detection, recognition, and forensic tests, extremely challenging, particularly when subjects are scanned from a remote.

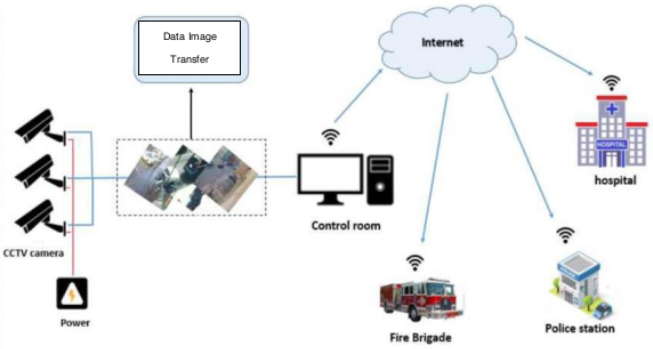

Figure 1. Framework of data transmission in CCTV system

The significance of the viewpoint is also required for CCTV system. The quality of the generated image is largely determined by the angle, defining the viability of the visual task, and simplifying its performance. As a consequence, intelligent CCTV sensors are required, from which recorded scene features can contribute to the recognition of unfamiliar objects by disambiguating between different interpretations. In short, visibility is essential for the sensor to recognize a feature [7].

Since there are several cameras, it would generates a substantial number of redundant and unvaluable image data, which causes problems such as looking for informative and valuable data from the stack of acquired data, as well as continual bandwidth loss [8].

Furthermore, image quality should be significantly increased when the relationship of illumination as well as the object area to be captured is thoroughly evaluated, to produce more informative photos, and making further processing much easier. Therefore, there is a need to useful data recognition and selection before passing it into the secure data transmission to control room. In this research, we provide a method for investigating image visibility scenarios with varying degrees of ambient lighting by considering previous research on face recognition.

There are two major processes in the course of face recognition: facial detection and facial identification. Among algorithms for facial detection, the most common being Haar-Cascade [9] and Histogram of Oriented Gradients (HOG) [10] classification methods. The

Haar-Cascade classifier is based on detecting Haar-like features. Using these as feature extractors with a multi-stage weak classification process (cascading), one can build a high-accuracy face classifier. The HOG algorithm divides image to small cells and calculates a histogram of gradient directions. With the fusion of grouped cells, the most frequent gradient direction in a block is kept. The resulting HOG descriptors are used to train a classifier, such as a Support Vector Machine (SVM), to detect faces.

Popular methods in facial identification include traditional machine learning, e.g., Eigenface [11], Fisherfaces [12] and local binary patterns histogram algorithm (LBPH) [13]. Another important model in facial identification is neural network model, in which deep neural network (DNN) [14] and convolutional neural network (CNN) [15] are widely adopted. OpenFace [16] is a DNN model for face recognition based on Google's FaceNet paper [17]. It is implemented through Python and Torch.

I. Mondal et al [18] proposed an electronic voting system for elections to ensure that each voter only casts once. Through the deep learning (CNN), the voter's face was recognized and verified. After the valid identity was confirmed, the vote could be made. The face data was deleted from the system after the vote process was completed.

L. Zhuang et al [19] proposed a new method based on deep learning to solve the adverse effects of environmental light changes in the process of face recognition. Firstly, a method of light preprocessing was applied to reduce the adverse effects of strong light changes on the face image. Secondly, the Log-Gabor filter was used to obtain Log-Gabor feature images of different sizes and directions, and then the Local Binary Pattern (LBP) was used to extract the features. Finally, the texture feature histogram was formed and inputted into the Deep Belief Network (DBN) for training.

J. Howse [20] used Haar cascade classifiers and methods such as LBPH, Fisherfaces, or Eigenfaces for object recognition, applied the detectors and classifiers on face recognition, and suggested the possibility of transferring to other fields of recognition. V. B. T. Shoba et al [21] proposed LBPH plus CNN's face recognition method, which not only reduced the computational cost, but also improved the face recognition accuracy to 98.6% in the Yale data set.

OpenFace is an open-source library that rivals the performance and accuracy of proprietary models. Another benefit of OpenFace is that the development of the model is mainly focused on real-time face recognition on mobile devices, so the model can be trained with high accuracy with very little data [16]. Because of these advantages, openface has been used in facial behavior analysis toolkit and get outstanding results [22].

Although face recognition technology has made great breakthroughs in recent years, it is still affected by different environmental lighting, resulting in a significant decline in accuracy and system failure. Thus, overcoming the adverse effect of ambient light is still one of the core problems of face recognition [23].

There are several reports or literatures investigate the effect of face recognition algorithms under different lighting conditions [24], but the detailed description about different lighting angles will affect the accuracy of programs is not mentioned. The LBPH algorithm has good robustness to extract important features of objects even with changes in environmental lighting. Here we train and test the LBPH and OpenFace face recognition models on Google's Colaboratory and compare the accuracy between two face recognition algorithms with changes in the degree of environmental lighting. Our findings are applicable to automated vision systems, which employ advanced algorithms to detect and track numerous pictures from multiple cameras. The analytical results can be utilized to recognize the object in a variety of lighting conditions.

2. Research Materials

2.1. Data set

The Extended Yale Face Database B data set [25] is a gray-scale face image dataset. Two data sets are provided:

Figure 2. Original image with azimuth angle of +130° to -130°

1. Figure 2 is the original image with an azimuth angle of +130° to -130°, the size is 640×480 pixels, and there are 28 people in files and 16,128 images in total. Each person has 9 poses, 64 images of environmental lighting changes, and a total of 576 images. 9 postures: posture 0 is the front posture, postures 1, 2, 3, 4, and 5 are about 12° from the camera's optical axis (distance from posture 0), and postures 6, 7 and 8 are about 24° [26].

Figure 3. Face image with an azimuth angle of +130° to -130°

2. Figure 3 shows face images with an azimuth angle of +130° to -130°, all manually aligned, cropped, and adjusted to 168×192 pixels. There are 38 people in total, and a total of 2,432 images. There is only one posture for each person, 64 images of environmental lighting changes, and a total of 64 personal images. In this paper we use these face images for face recognition [27].

64 kinds of ambient lighting are composed of different azimuth and elevation angles. The azimuth angle ranges from +130° to -130°. The elevation angle ranges from +090° to -040°. A positive azimuth angle indicates that the light source is on the right side of the object, while a negative azimuth angle indicates that the light source is on the left side of the object. A positive elevation angle means above the horizon, and a negative elevation angle below the horizon.

2.2. Google Colaboratory

Google's Colaboratory (Colab) provides an interactive environment that allows people to write and execute Python programs and Linux commands on the browser and use TPU and GPU at no cost. Therefore, Colab is suitable for training the small neural network model. In addition, Colab is not a static web page, but a "Colab notebook". One can edit the code, add comments, pictures, HTML, LaTeX and other formats, execute the code in sections, and save the current output results on Colab.

2.3. Python libraries

The face recognition algorithms and packages used in this paper are all written in the Python 3 programming language. Main libraries used are shown in Table 1:

Table 1. List of python libraries.

Library	purpose
opencv	image processing, LBPH face recognition
imutils	image processing (package opencv part of the API)
keras	data enhancement
numpy	array and matrix operations
sklearn	machine learning

pickle	object serialization and deserialization
sqlite3	database operations
matplotlib	visualized graphics presentation

The opencv and imutils packages are mainly used for image processing. The opencv functions include building LBPH face recognition models, reading images and caffe, executing tensorflow and other deep learning models, image preprocessing, capturing images from cameras, and displaying images. The imutils program is used in conjunction with opencv to calculate the number of image frames, resize the image, and maintain the image's aspect ratio. The keras library is used for data enhancement. It randomly performs operations such as horizontal translation, vertical translation, zooming, and horizontal flipping on the image to generate many similar images, increasing the number of images for training. The numpy library is used for one-dimensional array and matrix operations (images). The sklearn library is used to split the training set and test set, do label coding, execute model performance evaluation, and implement OpenFace's classifier—SVM. The pickle library is used to save the state of the object as a binary file for use in the next import program, training and test data, trained SVM classifier and label encoder (LabelEncoder). The sqlite3 library is used to store image paths, prediction results, and statistical results of training and testing. The matplotlib library is used to visualize the statistical results.

2.4. Data Transfer

Sending high-resolution multi-spectral pictures between a camera and an application can be particularly difficult, especially if transfer durations must be reduced or equal to image frame timings to avoid congestion in image data transmission [28].

Figure 4 describe a method to delivering pictures to clients over various transceiver connections that matches diverse transceiver technologies such as Ethernet, ESATA/SAS and PCI Express standards data is transmitted over the transceiver lines from the source, without any modifications in format or image processing.

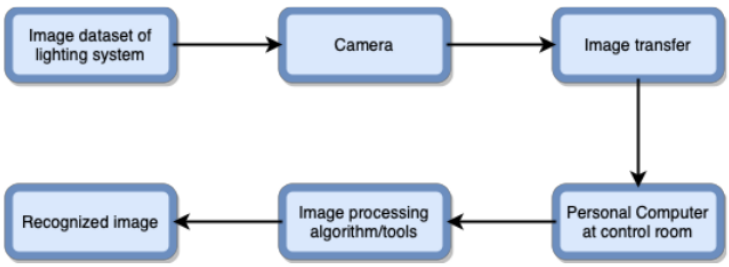

Figure 4. The process of image recognition from multiple cameras into PC

After receiving and transferring the image to a personal computer (PC) in the control room, the suggested approaches, LBPH and OpenFace are utillized to validate the images. In the next part, the image dataset was evaluated and successfully recognized the image's features.

3. Experiment Methods

3.1. LBPH: face recognition algorithm

LBPH is a method that can extract the texture features of the image through a local binary pattern (LBP) and conduct statistics through a series of histograms. Finally, after calculating the face distance by Euclidean distance, LBPH outputs the classification result.

3.2. OpenFace: face recognition algorithm

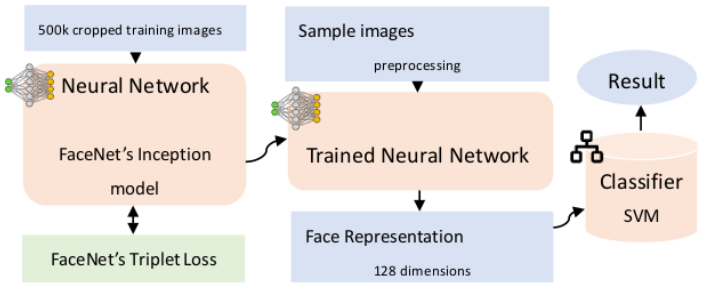

Figure 5. OpenFace nn4.small2 network structure

The model used in the research is OpenFace model nn4.small2.v1, and the network structure is shown in Figure 5. The accuracy of this model is 0.9292 ± 0.0134 [29], which is based on the Labeled Faces in the Wild (LFW) data set [30] as a benchmark. The Area Under Curve (AUC) is close to 1, which means the high accuracy of the model's prediction. The AUC of the nn4.small2.v1 model is 0.973.

3.3. Data preprocessing

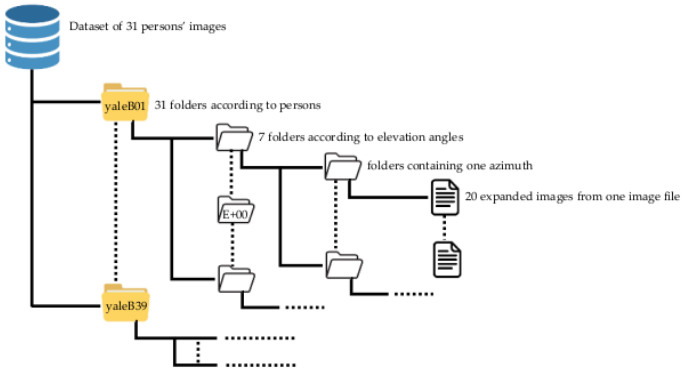

Figure 6. Group directory structure

First, download 38 person's pre-cut face images in The Extended Yale Face Database B data set from the website, and convert the images' format from PGM to JPEG for better image management. Among the downloaded images, 7 persons' images are partially damaged and removed, and the remaining 31 persons will be applied in the research.

Next, select one of 31 persons as the reference and 62 images (including duplicate images) from 64 images of the reference, divide these images into 7 groups according to the elevation angles (refer to Table 2), create 7 folders with the group name, and put images into the corresponding folder. Use the procedure to recur the remaining 30 persons with the same process as that of the reference person as shown in Figure 6. Use data enhancement technology for each image in each group by randomly panning, zooming, and flipping the images horizontally. Each image is expanded to 20 images and is stored in a folder with the same name as the image itself.

Finally, the 62 original images for each person are removed from the 7 group folders. Refer to Figure 7 for the complete process above.

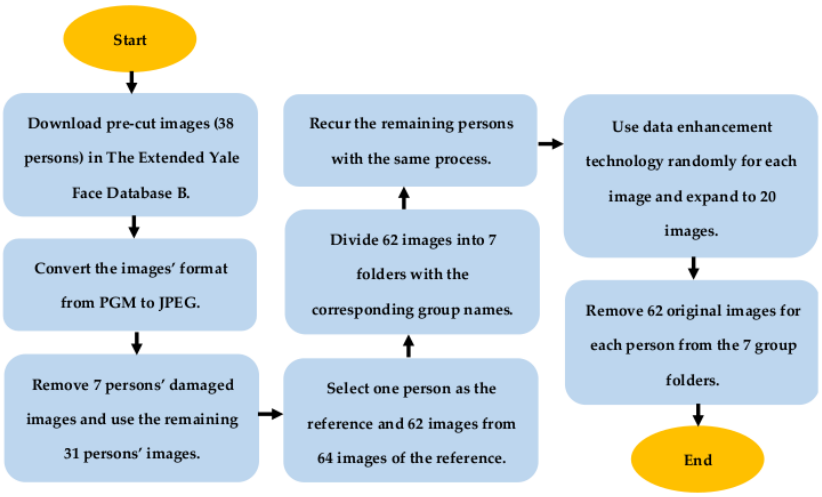

Figure 7. Data set processing flow chart

To briefly explain the rules of naming image names, the image name of yaleB01_P00A+000E+00 in Table 2 is used as an example: yaleB01 represents a person's name, and the three capital English letters P, A, and E represent Posture, Azimuth, and Elevation respectively. In this study only one posture existed in every person, and thus is recorded as P00.

Table 2. List of images grouped in one of 7-groups (group-A+120--120E+00)

Group name	Image name
A+120--120E+00	yaleB01_P00A+000E+00
	yaleB01_P00A+010E+00
	yaleB01_P00A+025E+00
	yaleB01_P00A+050E+00
	yaleB01_P00A+070E+00
	yaleB01_P00A+095E+00
	yaleB01_P00A+120E+00
	yaleB01_P00A-010E+00
	yaleB01_P00A-025E+00
	yaleB01_P00A-050E+00
	yaleB01_P00A-070E+00
	yaleB01_P00A-095E+00
	yaleB01_P00A-120E+00

3.4. Data set split

Table 3. Number of images in training set and test set after data enhancement

Item	Number of sheets	Number of images
Training set	31 people×62 sheets×20 sheets×0.8	30,752
Test set	31 people×62 sheets×20 sheets×0.2	7,688
Totally	31 people×62 sheets×20 sheets	38,440

As shown in Table 2, all the images after data enhancement are divided into two parts: 80% of the images are the training set, and 20% of the images the test set. However, it is different from the traditional splitting method. Traditionally, 80% of the images of everyone's images are added up to form the training set, and 20% of the images of everyone's images are added up to form the test set.

In this paper, the new splitting method is to read the 20 images after each data enhancement in each group of personal images and split each group into a training set and a test set. So that in each enhancement image group there are 16 training set images, and 4 test set images. The number of sheets in the training set and test set for each personal image is as the formulas (1) and (2).

$$\text{Training set: } 62 \text{ sheets} \times (20 \text{ sheets} \times 0.8) = 992 \text{ sheets} \quad (1)$$

$$\text{Test set: } 62 \text{ sheets} \times (20 \text{ sheets} \times 0.2) = 248 \text{ sheets} \quad (2)$$

Multiplying the formulas (1) and (2) by 31 (persons) is the number of sheets in the training set and test set as shown in Table 3. The reason for splitting immediately is that the prediction accuracy of each image needed to be counted later. If the traditional data splitting method is used, the data splitting will be uneven, and the subsequent statistics will lose accuracy.

3.5. LBPH model training

First, divide the image list of the data set (all image paths) into a training set and a test set, and store them in the Python list respectively. Next, extract the person's name from the path of each image as a label, save it in the Python list. Finally, serialize the image path and label of each test set into a binary file, and save it to the hard disk.

Read the images of the training set, convert them to grayscale images, and add them to the Python list. Through the label encoder, encode all tags into corresponding numbers which represent the same person. Then all gray-scale images and encoded labels are sent into the LBPH model for training. After the model is trained, save the LBPH model as a yaml format file, and save the label encoder as a binary file through serialization.

3.6. OpenFace model training

First, divide the image list of the data set (all image paths) into a training set and a test set, and extract the names of people from each image path as labels and store them in the Python list. Then store the image paths and tags of the training set and the test set into the SQLite database respectively.

Read the image of the training set, set the image size to a width of 600 pixels, and the height will automatically adjust with the width to maintain the aspect ratio. Use opencv's blobFromImage function to perform channel exchange, feature normalization and image size adjustment to 96×96 pixels. Send the image to the OpenFace pre-trained neural network nn4.small2.v1 model for inference, and output a 128-dimensional feature vector.

Use numpy's flatten function to flatten the 128-dimensional vector into a one-dimensional array and add the array to the Python list. After reading the images of the test set, skipping a flattened procedure, the images of the test set are processed in the same way as the training set, and finally stored in the Python.

While the labels are encoded in the same way as in LBPH, the 128-dimensional feature vector of the training set and the encoded labels are sent into the SVM classifier for training. After training, the SVM classifier, the label encoder, the 128-dimensional feature vector of the test set, and the names of the test set are individually stored as binary files through serialization.

3.7. LBPH prediction image

Load the trained LBPH model and label encoder from the hard disk and retrieve the image paths and tags of all test sets from the binary file. From the image path of each test

set, extract the person's name, group name, and file name. Read all the images and convert them into grayscale images and save them into the Python list. In the label part, the name of the person is label-encoded, and converted into the corresponding number. Send all grayscale images into the LBPH model for prediction. After the prediction is completed, save the group name, file name, test label, prediction label, and prediction distance into the Python list. Then use the Python dictionary to store the list with the name of the person as the key and the data as the value and save the dictionary as a binary file through serialization. Through deserialization, the binary dictionary file is read, and the person's name, group name, file name, test label, predicted label, and predicted distance in the list are extracted out from the dictionary by using the name of the person as the key. Then compare the test tags with the predicted tags one by one and record the identification results. At the end, the serial number, group name, file name, test label, predicted label, predicted distance, and identification result are stored in the SQLite database together.

3.8. OpenFace prediction image

Read the trained SVM classifier, label encoder, test set images, and labels from memory through deserialization. Encode the label through the label encoder and convert it into a corresponding number. Use the SVM classifier to predict each test set image and store the prediction results and probabilities. Get the image path of all test sets from the SQLite database and get the file name and group name from it. Compare the label of the test set with the predicted result and record the identification result. Finally, the serial number, person name, group name, file name, coded test set label, predicted label, predicted probability, and identification result are stored in the SQLite database.

3.9. Statistics and visualization

Obtain the prediction results from the SQLite database of LBPH and OpenFace respectively. Count the number of correct and incorrect predictions for each image of each group, and save the number, person name, group name, file name, status, and quantity into the SQLite database.

Figure 8. Grouped images of A+120--120E+00

For the visualization part, select the A+120--120E+00 grouping as shown in Figure 8 for more detailed statistics. This group consists of 13 photos, A+120--120 means the range is from +120° to -120° in azimuth, and E+00 means elevation is 0°. The reason for choosing this group is that the range of azimuth angles in this group is widely distributed, and E+00 makes illumination evenly in the longitudinal plane. With A+000 as the center, the lighting effect is symmetrical on the face, and is easier to observe at which azimuth will the accuracy rise or fall sharply.

4. Results and Discussion

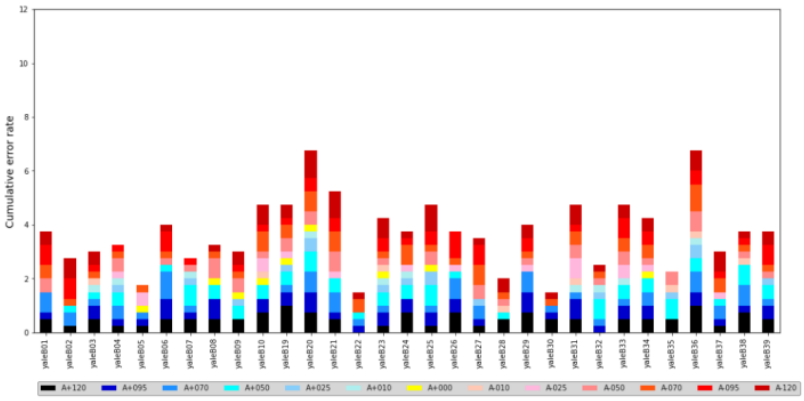

Figure 9. Comparison of the error rate of LBPH with different ambient light levels

Figure 9 is a stacking diagram of LBPH's A+120--120E+00 grouping error rate comparison of different ambient light levels. Figure 10 illustrates the stacking method of Figure 9. The bottom of the stacked image is the azimuth +120°, the center white block the azimuth +000°, the top the azimuth -120°. The lower recognition error rate while closer to the center white block, and otherwise higher. Therefore, the recognition error rate of the uppermost and the lowermost blocks are the highest, and the recognition error rate decreases as the azimuth angle approaches +000°.

Figure 10. Illustration of stacked graphs of different illumination error rates for a single user

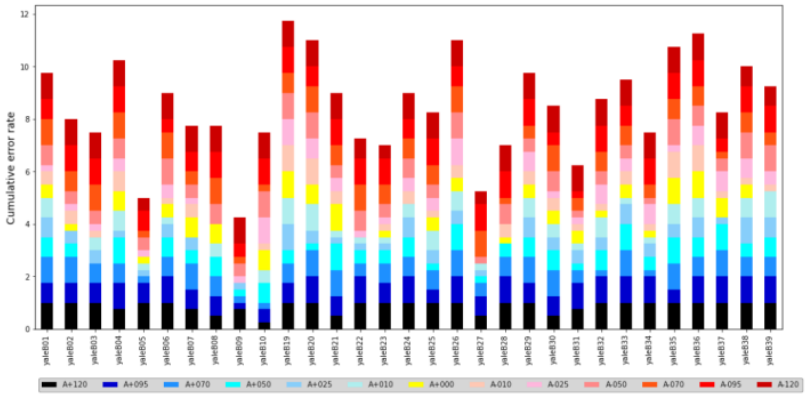

Figure 11. Comparison of error rate of OpenFace with different ambient light levels

Figure 11 is a stacking diagram of OpenFace's A+120--120E+00 grouping error rate comparison of different ambient light levels. The recognition error rates of the uppermost and the lowermost blocks are the highest and decrease as the azimuth angle approaches +000°. Overall, OpenFace's recognition error rate is about 20% to 49% higher than LBPH's in A+120--120E+00 grouped images.

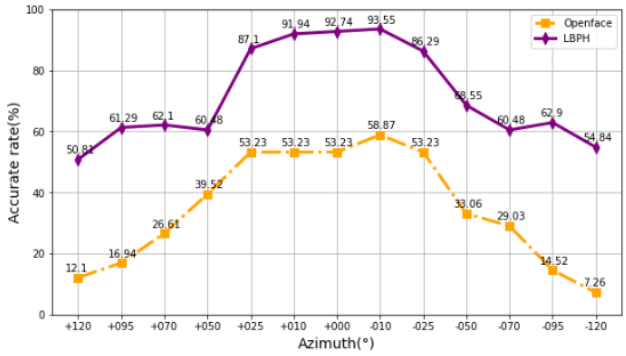

Figure 12. Average ambient lighting accurate rates of A+120--120E+00 group

$$\text{Accurate rate} = \sum_{1}^{31} ((\text{Number of correctly identified sheets} \div 4) \times 100\%) \div 31 \quad (3)$$

Figure 12 shows the average ambient lighting accurate rates of 31 people grouped by A+120--120E+00. Formula (3) is the calculation formula for the average recognition accurate rates of each azimuth angle in Figure 12.

Firstly, calculate the ratio of the number of correct predictions to the total number of predictions, then convert the ratio into a percentage. Finally, sum up all the percentages, then divide by 31 people to get the average.

For the picture with A+000° as the center line, the recognition accurate rate is almost symmetrical. There is not much difference between the light source on the left or the right. The closer the recognized azimuth angle is to azimuth +000°, the higher the recognition accurate rate; the farther away the azimuth angle +000°, the lower the recognition accurate rate. Overall, the recognition accurate rate of LBPH under changes in environmental lighting is far better than that of OpenFace. Therefore, LBPH is more suitable for applications in environments with light changes. From the figure, whether it is LBPH or

OpenFace at the azimuth angle of -010°, the recognition accurate rate is the highest. LBPH is 34.68% more accurate than OpenFace at the azimuth angles of -010°. LBPH is 38.71% more accurate than OpenFace at the azimuth angles of +120°.

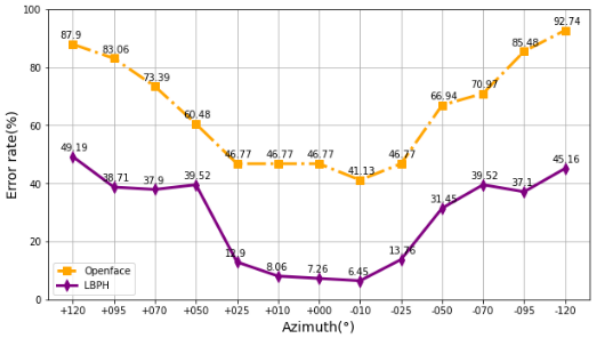

Figure 13. Average ambient light error rate of A+120--120E+00 group

$$\text{Error rate}=\sum_{1}^{31}(\text{Number of sheets with identification errors} \div 4)\times 100\%) \div 31 \quad (4)$$

Figure 13 shows the average ambient light error rate of 31 persons grouped by A+120--120E+00, and formula (4) is the calculation formula for the average identification error rate of each azimuth angle shown in Figure 13. First, calculate the ratio of the number of prediction errors to the total number of predictions, and convert it into a percentage. Finally sum up all percentages and divide by 31 people to get the average.

With A+000° as the center line, the recognition error rate is almost symmetrical. Overall, the overall recognition error rate of LBPH is much lower than that of OpenFace. When both azimuth angles are in the range of +25° to -25°, the recognition error rate is relatively low. When both azimuth angles are in the range of +50° to +120° and -50° to -120°, the recognition error rate is relatively high. For LBPH, when the azimuth angle is +50° to +25°, the identification error rate is reduced by 26.62%; when the azimuth angle is -50° to -25°, the identification error rate is reduced by 17.74%. For OpenFace, when the azimuth angle is +50° to +25°, the recognition error rate drops by 13.71%; when the azimuth angle is -50° to -25°, the recognition error rate drops by 20.17%. From the azimuth angle of +50° to +25° and -50° to -25°, it is found that the recognition error rate is significantly reduced.

It means that the change of the azimuth angle reduces the shadow on the face, and the contours of the facial features are clearer, reducing the error rate of face recognition.

5. Conclusions and Future Work

From the results, LBPH is more suitable than OpenFace in recognition applications with ambient lighting changes. When the light source is kept larger than the azimuth angle of +25° or less than -25° and the elevation angle +000°, the shadows on the face will increase and the recognition accuracy will be lower, otherwise the result will reverse. According to the results of face recognition with changes in ambient light, compared to the OpenFace recognition model, LBPH has superior performance in classification and recognition.

Therefore, for image recognition (such as face recognition) that requires more detailed output for line texture features, but is interfered by ambient light, LBPH will have a greater level of recognition accuracy for its object recognition or classification results compared to OpenFace. Furthermore, the previously made important image collection can be employed in the subsequent data transfer process.

Since the trained LBPH algorithm is very robust to changes in ambient light, it can be deployed on a Raspberry Pi or other edge computing processors to be implemented in access control systems with ambient lighting changes or related fields of applications, such as facial payment systems.

References

[1] M. Abd Al Rahman and A. Mousavi, "A review and analysis of automatic optical inspection and quality monitoring methods in electronics industry," *IEEE Access*, vol. 8, pp. 183192-183271, 2020, doi: 10.1109/ACCESS.2020.3029127.

[2] D. Salama AbdELminaam, A. M. Almansori, M. Taha, and E. Badr, "A deep facial recognition system using computational intelligent algorithms," *Plos one*, vol. 15, no. 12, p. e0242269, 2020, doi: https://doi.org/10.1371/journal.pone.0242269.

[3] Y. Adini, Y. Moses, and S. Ullman, "Face recognition: The problem of compensating for changes in illumination direction," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 7, pp. 721-732, 1997.

[4] H. Alshamsi, H. Meng, and M. Li, "Real time facial expression recognition app development on mobile phones," in *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, 2016: IEEE, pp. 1750-1755.

[5] T. Huang, Z. Xiong, and Z. Zhang, "Face recognition applications," in *Handbook of Face Recognition*: Springer, 2005, pp. 371-390.

[6] F. W. Wheeler, R. L. Weiss, and P. H. Tu, "Face recognition at a distance system for surveillance applications," in *2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 2010: IEEE, pp. 1-8.

[7] A. Mittal and L. S. Davis, "Visibility analysis and sensor planning in dynamic environments," in *European conference on computer vision*, 2004: Springer, pp. 175-189.

[8] K. M. Ijaz Ul Haq, M. Sajjad, M. Y. Lee, D. Han, and S. W. Baik, "A Study of Data Dissemination in CCTV Surveillance Systems," *image*, vol. 75, pp. 14867-14893, 2016.

[9] S. Soo, "Object detection using Haar-cascade Classifier," *Institute of Computer Science, University of Tartu*, vol. 2, no. 3, pp. 1-12, 2014.

[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, 2005, vol. 1: Ieee, pp. 886-893.

[11] J. Zhang, Y. Yan, and M. Lades, "Face recognition: eigenface, elastic matching, and neural nets," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1423-1435, 1997.

[12] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 7, pp. 711-720, 1997.

[13] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *European conference on computer vision*, 2004: Springer, pp. 469-481.

[14] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11-26, 2017.

[15] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE transactions on neural networks*, vol. 8, no. 1, pp. 98-113, 1997.

[16] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "Openface: A general-purpose face recognition library with mobile applications," *CMU School of Computer Science*, vol. 6, no. 2, p. 20, 2016.

[17] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815-823.

[18] I. Mondal and S. Chatterjee, "Secure and hassle-free EVM through deep learning based face recognition," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 2019: IEEE, pp. 109-113.

[19] L. Zhuang and Y. Guan, "Deep learning for face recognition under complex illumination conditions based on log-gabor and LBP," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2019: IEEE, pp. 1926-1930.

[20] J. Howse, "Training detectors and recognizers in Python and OpenCV," in *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2014: IEEE Computer Society, pp. 1-2.

[21] V. B. T. Shoba and I. S. Sam, "Face recognition using LBPH descriptor and convolution neural network," in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2018: IEEE, pp. 1439-1444.

[22] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency, "Openface 2.0: Facial behavior analysis toolkit," in *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, 2018: IEEE, pp. 59-66.

[23] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM computing surveys (CSUR)*, vol. 35, no. 4, pp. 399-458, 2003.

[24] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE transactions on image processing*, vol. 19, no. 6, pp. 1635-1650, 2010.

[25] U. o. C. S. Diego. "The Extended Yale Face Database B." http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html (accessed 2020/09/30.

[26] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE transactions on pattern analysis and machine intelligence*, vol. 23, no. 6, pp. 643-660, 2001.

[27] K.-C. Lee, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 5, pp. 684-698, 2005.

[28] P. N. Zanjani, M. Bahadori, and M. Hashemi, "Monitoring and remote sensing of the street lighting system using computer vision and image processing techniques for the purpose of mechanized blackouts (development phase)," 2013.

[29] OpenFace. "Models and Accuracies." https://cmusatyalab.github.io/openface/models-and-accuracies/ (accessed 2022/09/18.

[30] U. o. Massachusetts. "Labeled Faces in the Wild." http://vis-www.cs.umass.edu/lfw/ (accessed 2022/09/18.

An Enhanced LBPH Approach for Face Recognition Affected 2 by Ambient Light in Data Transmission Planning

ORIGINALITY REPORT

13%	9%	8%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Loughborough University Student Paper	2%
2	etds.lib.ncku.edu.tw Internet Source	1%
3	Submitted to The University of Manchester Student Paper	1%
4	Liyun Zhuang, Yepeng Guan. "Deep Learning for Face Recognition under Complex Illumination Conditions Based on Log-Gabor and LBP", 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2019 Publication	1%
5	eprints.qut.edu.au Internet Source	1%
6	Bilel Ameur, Mebarka Belahcene, Sabeur Masmoudi, Ahmed Ben Hamida. "Unconstrained Face Verification Based on Monogenic Binary Pattern and Convolutional	1%

Neural Network", 2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), 2020
Publication

7	Mohamed Amine Hmani, Dijana Petrovska-Delacretaz. "State-of-the-art face recognition performance using publicly available software and datasets", 2018 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), 2018 Publication	<1 %
8	Submitted to University of Central Lancashire Student Paper	<1 %
9	Stan Li. "", IEEE Transactions on Pattern Analysis and Machine Intelligence, 4/2007 Publication	<1 %
10	Zhijie Zhang, Jianmin Zheng, Nadia Magnenat Thalmann. "Engagement Intention Estimation in Multiparty Human-Robot Interaction", 2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN), 2021 Publication	<1 %
11	ijarcsee.org Internet Source	<1 %
12	www.set-science.com Internet Source	<1 %

13	api.intechopen.com Internet Source	<1 %
14	research-repository.griffith.edu.au Internet Source	<1 %
15	Zhan-Li Sun, Li Shang. "A local spectral feature based face recognition approach for the one-sample-per-person problem", Neurocomputing, 2016 Publication	<1 %
16	C. Piciarelli, C. Micheloni, G.L. Foresti. "PTZ camera network reconfiguration", 2009 Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC), 2009 Publication	<1 %
17	Xiao Jian Tan, Wai Zhe Leow, Wai Loon Cheor. "A Simple, Precise, and High-Speed Die Edge Detection Framework Based on Improved K-Mean and Landscape Analysis for the Semiconductor Industry", Arabian Journal for Science and Engineering, 2021 Publication	<1 %
18	biblio.ugent.be Internet Source	<1 %
19	Ahamed Yasar Z., Dinesh Kumar R., S. Aadarsh, Hari Kumar G.. "Border Surveillance System using Computer Vision", 2020 6th International Conference on Advanced	<1 %

Computing and Communication Systems (ICACCS), 2020
Publication

20	www.biorxiv.org Internet Source	<1%
21	annals-csis.org Internet Source	<1%
22	Josiah-David Sykes, Randy St. Fleur, Daler Norkulov, Ziqian Dong, Reza K. Amineh. "Conscious GPS: A System to Aid the Visually Impaired to Navigate Public Transportation", 2019 IEEE 40th Sarnoff Symposium, 2019 Publication	<1%
23	Md Shamim Hossain. "Microc alcification Segmentation Using Modified U-net Segmentation Network from Mammogram Images", Journal of King Saud University - Computer and Information Sciences, 2019 Publication	<1%
24	Orioli, Aldo, and Alessandra Di Gangi. "An improved photographic method to estimate the shading effect of obstructions", Solar Energy, 2012. Publication	<1%
25	Pradeep Kumara V. H, Ravindra P. Rajput. "Design and implementation of convenient and compulsory voting system using finger	<1%

print sensor and GSM technologies", 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), 2020
Publication

26	Submitted to University of Limerick Student Paper	<1%
27	"Intelligent Cyber-Physical Systems for Autonomous Transportation", Springer Science and Business Media LLC, 2022 Publication	<1%
28	api.openalex.org Internet Source	<1%
29	cs229.stanford.edu Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		