Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7766343

ILLUMINATION FACE RECOGNITION USING DEEP LEARNING TECHNIQUES

Bharath J[1]
Research Scholar, Department of CSE, Rajarajeshwari College of Engineering, Bengaluru

Dr. R.Balakrishna[2]
Department of CSE, Rajarajeshwari College of Engineering, Bengaluru

Abstract
Within the realm of biometrics, facial recognition methods have emerged as one of the most promising areas of study in the previous decade. Lighting conditions that cannot be adjusted provide a significant obstacle to accurate face recognition, despite receiving a great deal of attention and study. In this research work, we recommend a facial recognition technique under illumination condition. The technique consists several stages in it. The strategies for face identification under variable lighting circumstances that are detailed in this work are broken down into a few different steps. Face detection using MTCNN, data normalization and pre-processing with random Gaussian noise, model development for face categorization, and authentication method are the key phases in this proposed technique.
Keywords: Face Recognition, MTCNN, Variant Illumination, Random Gaussian Noise

1. Introduction

Face recognition is a biometric practice for verifying the identity of an individual based on a video or photograph of that person [1]. There are several applications where it has proven useful. Facial recognition has many practical uses, including in the fields of virtual reality, retail, promotional activity, games, cybersecurity, forensics, teleconferencing, smart meetings, visual monitoring, and counter-terrorism [2]. What this means is that each person's appearance is compared to a database of preexisting face images to monitor which ones are most similar [3].

Face identification is a highly difficult study field since even photographs of the same person might appear different owing to alterations in occlusion, lighting, expression, and stance [4]. A lot of recent work has been done on face recognition, and some of it has shown that lighting conditions can affect how well certain face recognition algorithms work [5]. Conversely, training or testing any image data is sensitive, even under fluctuating levels of light. These aspects contribute to the difficulty of face recognition and have received a lot of study over the past several decades [6]. In order to account for the varying levels of light, a great many different algorithms have been suggested.

One of these methods makes use of image processing modelling approaches, which are effective when attempting to normalize faces that have been exposed to a variety of lighting conditions. Histogram equalization (HE) [7], Gamma intensity correlation [8], or logarithm transformations [9] are all effective methods for achieving this goal, and each one excels in a unique set of circumstances. To eliminate the impact of lighting on faces, several suggested models are currently in widespread use.

Tumblin *et al.* attempted to eliminate lighting issues in images using a low curvature image simplifier (LCIS) with contrast enhancement [10]. However, this method's complexity is raised by the necessity of human intervention in the selection of fewer than eight different factors. Even when employing these global processing approaches, dealing with unevenly varying light is still too difficult. Sometimes, localized histogram equalization (RHE) [8] and block-based histogram equalization (BHE) [11] are used to deal with lighting inconsistencies.

In this article, we present a method of face recognition that works in different illumination environments. There are several phases to the procedure. Several distinct ways are outlined in this paper for recognizing faces in varied lighting circumstances. Section II of this study presents works related to this study. Section III describes the detailed methodology, equations, and proper figures required in this work. The next section discusses results obtained in different stages. The final section presents the conclusion of this article.

2. Related Works

The last ten years have seen a plethora of studies devoted to improving face recognition algorithms, with methods like linear discriminant analysis (LDA) and deep neural networks (DNNs) among the most prominent (LBP). In optimal settings, these techniques have shown promising outcomes. However, these techniques failed miserably when put to the test by factors such as fluctuating lighting and facial expressions. Several issues in photographic data processing, including entity identification, feature mining, and picture classification, have been successfully tackled thanks to CNN-based approaches. Some researchers have proposed using a Convolution Neural Network to speed up the facial recognition system, which is why this technique is being promoted as an alternative to more conventional approaches. This part will discuss the recent groundbreaking work done on FR using convolutional neural networks CNNs.

K. Simonyan *et al.* (2014) employed a VGG-16 design for huge-scale picture categorization [12]. During this process, they demonstrated that depth is beneficial for improving the accuracy of classification. This method has been successful in achieving 98.95% and 97.3% accuracy, respectively, when employed to the Labeled faces in the wild dataset (LFW) and YouTube Faces (YTF) that were used in the study by Parkhi *et al.* in 2015 [13]. In this study, the authors also used the architecture that was proposed in [5]. Chen *et al.* [14] developed a technique in 2016 which was based on the DCNN network was based-on [15]. It obtained 97.45% accuracy on the (LFW) dataset.

Later, the Facial Attribute Assistant Network (FAAN) was presented in 2017 by Chan *et al.* [16] and was constructed on the Residual Network ResNet-101 that was completed on IJB-A. This network has reached 98.2%. A technique based on the AlexNet model and the Inception-ResNet-V1 model was developed in 2018 by A. Zhanfu *et al.* and analyzed on the LFW and SLLFW datasets [17]. Correspondingly, the Inception-ResNet-V1 network obtained a recognition rate of 99.20% on LFW and 95.80% on SLLFW.

3. Methodology

The face recognition under variant illumination condition techniques described in this study has several stages in it. The stages are mainly: face detection using MTCNN, data normalization and pre-processing using random Gaussian noise, model design for face classification, and authentication mechanism. The flowchart presented in figure 1 depicts the workflow of this study methodology.

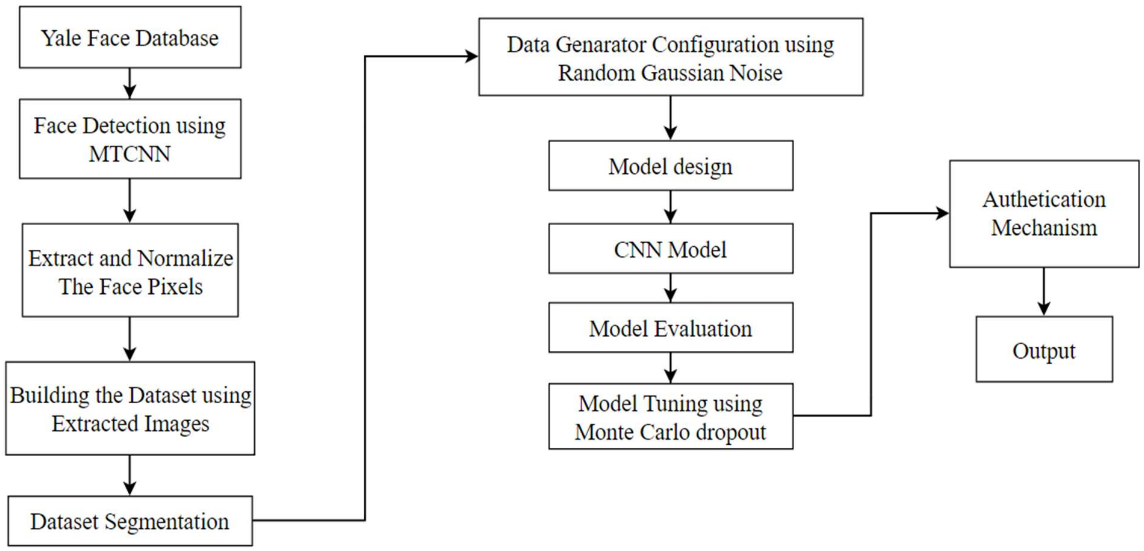

Fig. 1. Work Flow of This Whole Experimental Study Methodology

3.1. Input Dataset

In this experimental study, we utilized the Yale face database [18]. There are 165 photos of 15 different persons in this database. Each person had eleven pictures taken of them, one for each possible lighting or emotion combination: normal, right-light, sad, drowsy, astonished, and wink; centre-light; with glasses; joyful; left-light; without glasses; and normal. This database is proved to be quite helpful when initiating tests relating to facial recognition.

3.2. Face Detection Using MTCNN

However, the efficiency of facial recognition technology is further diminished by the presence of divergent expressions under varying lighting conditions, which is currently an unresolved problem [19]. In each of the pictures, the subject's face may be seen in a different part of the frame. The first thing that needs to be done is to standardize all of the face locations so that they can be input into a classifier in a reliable manner. There are a few different approaches one might use to accomplish this goal. Haar Cascades offer a straightforward and speedy approach to the detection of faces, but the results they produce are not always accurate [20]. They perform quite well when used for detection in real-time. When it comes to facial recognition, you have the luxury of taking your time with the processing of the image. As a result, we made use of MTCNN [21], an algorithm for identifying faces. This algorithm employs a CNN architecture, which, in comparison to previous methods, results in significantly greater accuracy.

The detected data using this algorithm is provided as a collection of JSON objects. There are three primary keys in every JSON entity, and they are named "container," "certainty," and "descriptor," respectively. The box's dimensions are listed with [x, y, width, height] in the 'box' key. The certainty indicates how likely it is that the face is included inside the bounding box. The data is stored as a JSON object through the following keys: left eye, right eye, nose, left-mouth, and right-mouth. One pixel coordinate identifies each landmark (x, y). The next step is to extract and normalize the face pixels so that they can be reliably used for classification. We converted the images to a NumPy array, resized them and then converted to grey images to employ the MTCNN algorithm to them to detect faces [22].

3.3. Dataset Building and Data Segmentation

At first, we extracted faces from all the images present in the database using the MTCNN method. After that, we created a directory to save those extracted faces. Figure 2 shows some samples of those extracted images we saved in the directory.

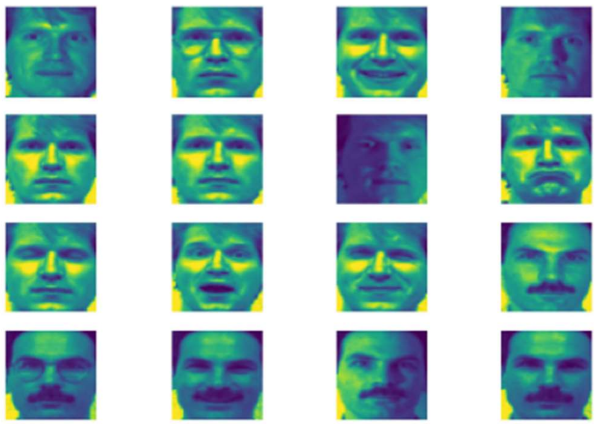

Fig. 2. Extracted Faces in the Created Directory

After the face directory is set up, we divided the database directory into train and testing sets. In this process, we dropped the subjects (if any) who had any coerce errors. There are a limited number of samples per class in this case, although we need to have a suitable number of classes to benchmark appropriately. We decided to segment the dataset in a 7:3 ratio for training and test datasets. In this way, we can get three images for each test class.

3.4. Random Gaussian Noise

Noise is typically defined as a series of unpredictable variations in the luminance or colour of the picture. The probability density function of Gaussian Noise is the same as that of the normally distributed function. Noise like this is generated by augmenting the Image function with a random Gaussian function. We created a random Gaussian noise function with a variability score of 35 to add it to the face images using the 'image data generator' function from TensorFlow Keras [23].

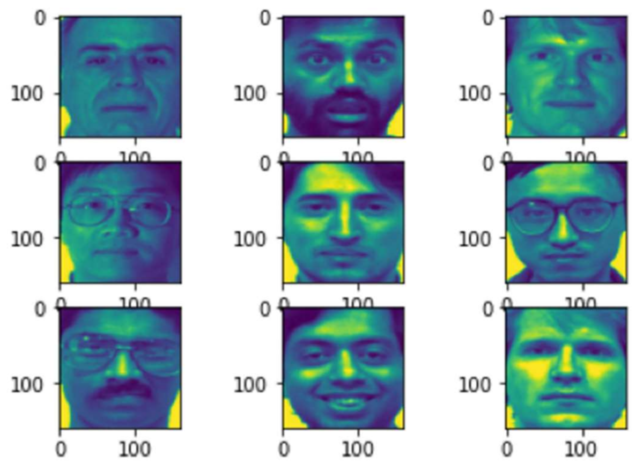

Fig. 3. Results from Data Generator Configuration Using Random Gaussian Noise

In this way, the dataset can be augmented as they have a limited amount of data for training and test subsets, along with making the model robust towards similar noises. If p(x,y) depicts an image function and q(x,y) is the Gaussian noise, then the resultant image after adding the noise to the image is presented as:

$$\text{Resultant image} = R(x, y) = p(x, y) + q(x, y) \tag{1}$$

The following equation can be used to determine a Gaussian random variable's probability density function, denoted by $p_{Gaussian}$. In this equation, z denotes the greyscale level, u stands for the overall average and σ is the standard deviation.

$$p_{Gaussian}(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-u)^2}{2\sigma^2}} \tag{2}$$

3.5. Label Encoding

We employed label encoding as a required pre-processing step here. In order to make labels understandable by computers, a method known as 'label encoding' is used. Sklearn is a powerful tool for quantitatively representing the degrees of classified characteristics. When using 'LabelEncoder', categories can have a value between 0 to n classes-1, where n is the cumulative number of categories.

3.6. Model Design for Classification

We employed different models for the dataset to classify the data into expected classes. We started with the benchmark models: SVM and random forest methods. After that, we built a CNN model and tuned its performance using the Monte Carlo dropout method.

3.6.1. SVM Model

A supervised machine learning method, Support Vector Machine (SVM), can be used for classification and regression. The target of the SVM technique is to trace a hyperplane in an N-dimensional environment which can be used to categorize the input points reliably. A SVM model provides us with a solid estimate of the dataset's potential performance. We utilized

SVC or support vector classifier to fit the model. We employed this classifier to train and test datasets to determine model performance.

3.6.2. Random Forest Model

The idea of ensemble methods, in which several classifiers are combined to tackle a difficult issue and boost the effectiveness of the model, forms the basis of Random Forest, a widely used machine learning technique. High precision and immunity to overfitting are two benefits of a more extensive tree forest. We used the random forest classifier in this study for the category prediction of sentiment classes.

3.6.3. CNN Model

In deep learning, CNN or convolutional neural network is a technique of neural networks used mainly for grid-like topology-based data such as an image. Convolution is defined as the process of multiplying two functions point-by-point to generate a third function. In this case of sentiment classification based on image data, we have a filter and a matrix of picture pixels. We calculate the dot product of the two matrices by sliding the filter across the picture. A feature map describes the resultant matrix.

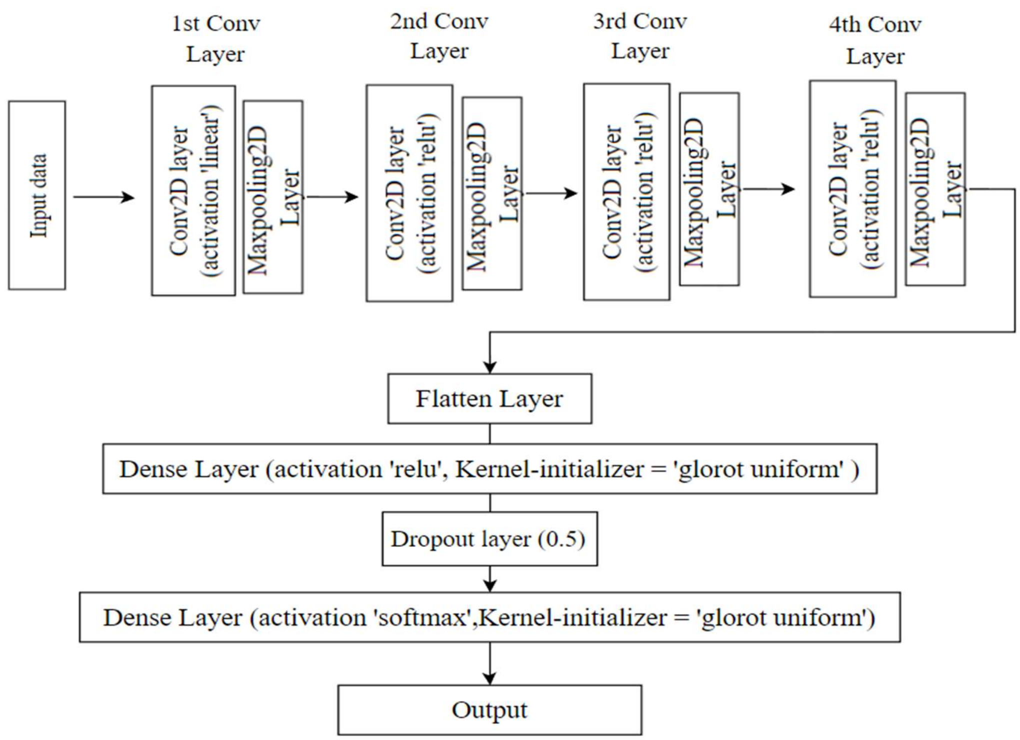

Fig. 4. CNN Model Architecture

In this study, we have four such convolutional layers present in the intended CNN model. Each of these layers has a conv2D layer and a 2×2 max-pooling layer for feature extraction. We fed the input data (pre-processed and label-encoded) to the first convolutional layer to create a 3×3 sized kernel with 32 filters. After the 4^{Th} convolutional layer, the data shape is flattened for classification purposes. Then, we employed the flattened data to a dense layer with 512 neurons. Every neuron in a dense layer receives an input from the layer below it and generates an output for the layer above it. Dropout is a technique to tackle overfitting and improve neural network generalization. In this case of the CNN model, we added a dropout layer with 50%

dropout. The greatest amount of variation in this distributed population may be obtained by discarding a neuron with a likelihood of 0.5. Finally, the dense layer with 15 neurons is used for the classification task.

We used categorical cross-entropy for loss calculation in this case. The learning rate was set to be 0.0003. The model was trained for 25 epochs.

3.7. Monte Carlo Dropout

Monte Carlo Dropout, suggested by Gal and Ghahramani (2016) [24], is a brilliant insight that the application of the standard dropout may be viewed as a Bayesian estimate of an established prediction measure: the Gaussian method. These several networks with varying numbers of neurons removed can be viewed as Monte Carlo sampling from the whole set of models. Mathematically, the predictive Gaussian distribution can be described as follows.

$$p(y \mid x, \Theta) = \mathrm{N}(f(x, \Theta), s^2(x, \Theta)) \qquad (3)$$

where, N indicates the Gaussian function and f(x, Θ) indicates the average, when $s^2(x, \Theta)$ is the variance. These parameters are the outputs from the Monte Carlo dropout Bayesian Neural Network.

This not only enhances the model's performance but also gives a mathematical foundation on which to discourse about its ambiguity. This dropout algorithm relies on the random inhibition of neurons to achieve its goals.

3.8. Authentication Mechanism

After all these procedures, we created an authentication mechanism to apply for this whole work practically. We can filter against this technique by ensuring that a person is not allowed into the building if the classification is below a certain threshold. This threshold is set at 80. The mechanism is set up along with a list of persons who can be permitted inside the system. At this stage, we created a list of eight classes which can be granted permission. These classes are those which are labelled as 1, 3, 5, 7, 9, 11, 13, and 15. The output shows up with the 'Welcome', 'face not recognized', and 'Not permitted' messages against the corresponding inputs.

So, the proposed algorithm is presented below.

The Proposed Algorithm for Face Reorganization Under Illumination Condition

While (Input! =0) {

START:

Stage 1: Input image files are converted into array.

The sequences of image pixel arrays are joined in a stack.

Stage 2: Applied MTCNN to the entire dataset.

Detect faces in images with bounding boxes.

Stage 3: Extract the bounded faces in different file.

Save the image files in a different directory.

Stage 4: Dataset segmentation into train and test sets

Stage 5: Random Gaussian noise applied to the image data to avoid overfitting.

Stage 6: Design CNN model.

Stage 7: Apply Monte Carlo dropout.

Stage 8: Create and apply an authentication mechanism.

Stage 9: Show output and related information

END.

4. Results And Discussion

Figure 5 shows the outputs from the face detection and extraction step of this study. It depicts a sample input RGB image data with a shape of (243, 320, 3) normalized and reshaped. Then it is fed to the MTCNN face extractor model to detect the face in the image. The MTCNN model detects the existing face in the image with a bounding box. After that, the detected face is extracted and normalized for further experimental study.

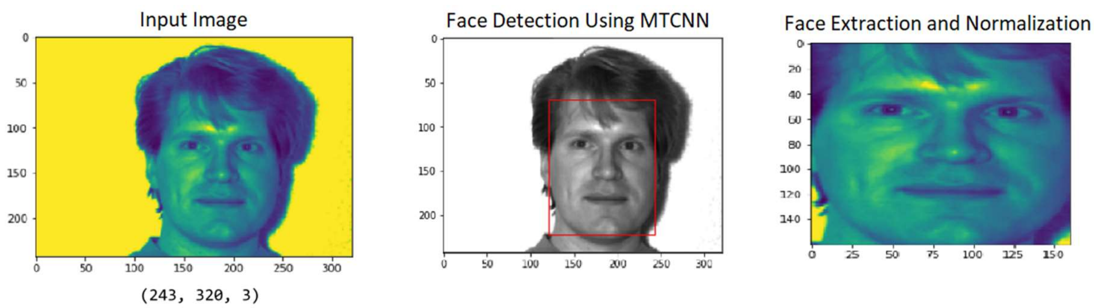

Fig. 5. Face Detection Result Using MTCNN and Face Extraction for Model Implementation

After the model fitting to the data subsets using the SVM classifier, we found that it reached a 94.16% accuracy during training. Still, when employing this model in the test dataset, it showed an accuracy of 60%. The precision, F1-score, and recall are high in the cases of categories 4, 6, 7, 10, 11, and 13. The model can only predict 2 out of 3 samples for the rest of the categories.

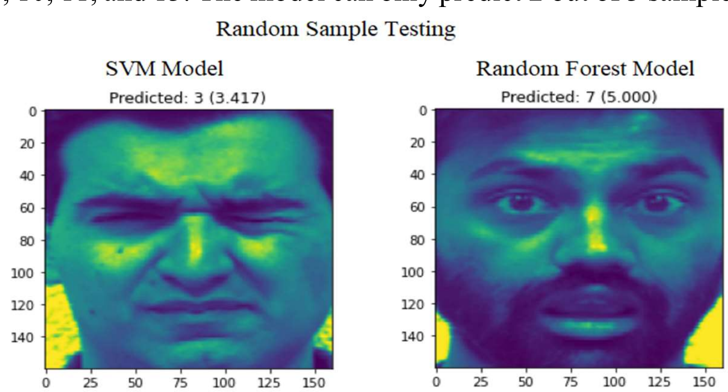

Fig. 6. Random Sample Testing Results Using SVM and Random Forest Model

The test dataset reached an accuracy of 80% when employed in the random forest classification. The outcomes show that the precision, recall and f1 scores in the case of classes 4 to 11, along with class 2 is, high, and the rest is not getting predicted accurately enough. This outcomes depicts that the random forest classifier performs better than the SVM classifier. Figure 6 shows random sample testing results from the SVM and random forest models. In both the test cases, both of these models predicted the correct classes.

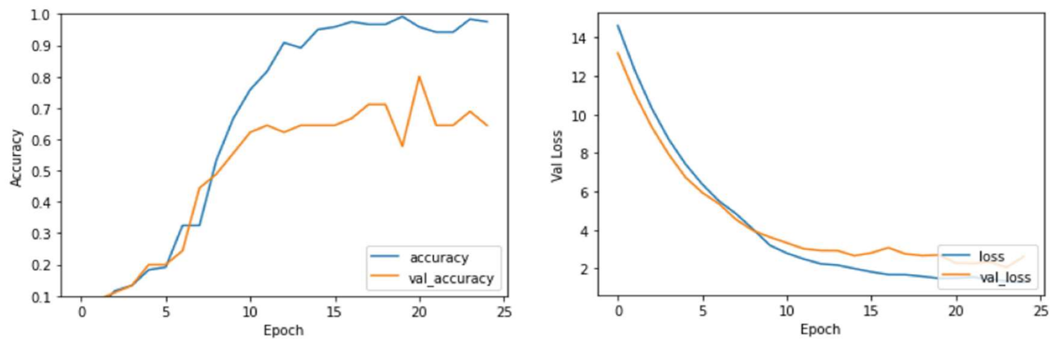

Fig. 7. CNN Model Evaluation Graphs as Epochs Change – (a) Accuracy and Validation Accuracy Comparison; (b) Loss and Validation Loss Comparison.

Figure 7 shows the evaluation graphs created using the output values obtained in each epoch from the CNN model implementation. The graph depicts that the loss values in the training and validation stages are almost saturated, which is suitable for the model assessment, but the training and validation accuracy go far apart from each other after the 8th epoch. This is not a good sign to use the CNN technique without any further tuning. That is why we employed the Monte Carlo dropout technique for the CNN model to perform efficiently.

Table 1: Results Obtained from different models.

Models	**Accuracy**	**Classes Recognized Accurately**
SVM	60%	6
Random forest	80%	9
CNN without Monte Carlo Dropout	62%	5

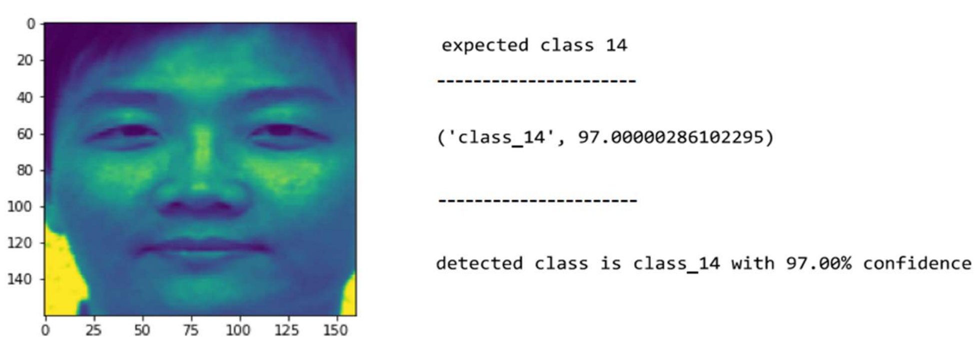

Fig. 8. Random Sample Testing Output After Applying Monte Carlo Dropout

If there isn't enough information to go on, or if the network is too complicated, the model may simply memorize the training data and perform well only on the data it has seen before, failing miserably when presented with novel data. Overfitting describes this situation. Overfitting is avoided with the aid of dropout. Monte Carlo dropout methods are a category of statistical computing techniques that generate distributions of numerical quantities by iterative random sampling. Using the Monte Carlo method, the CNN technique performs more efficiently with the test dataset. The sample test shown in figure 8 presents the expected class and the predicted class accurately with 97% confidence.

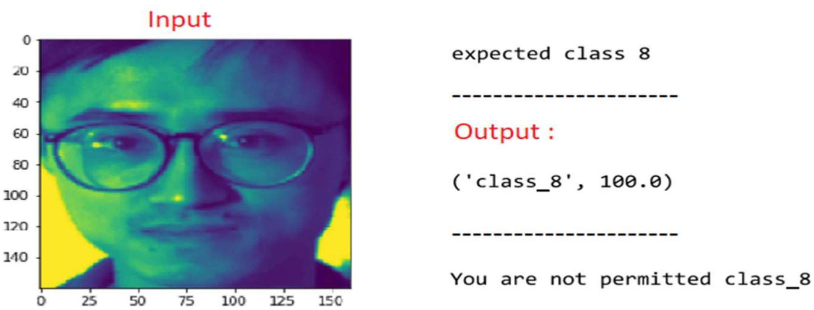

Fig. 9. Random Sample Testing for the Authentication Mechanism

Figure 9 presents a random sample test for the proposed authentication mechanism using the CNN with the Monte Carlo dropout method. It depicts an input image and its expected class. The output shows the predicted class after the reorganization step of the face is done. It also indicates the authentication of the input image and if the person is permitted or not. As the predicted class is 'class-08', which is absent from the list of authorized persons, the mechanism does not allow the system access to the person.

5. Conclusion

For the past decade, research into face recognition techniques has become one of the most exciting developments in the domain of biometrics. While much research has been done to improve the accuracy of face recognition, one major issue remains uncontrollable lighting. This research proposes a face recognition technology that works in different illumination environments. There are several phases to the procedure. Several distinct ways are outlined in this paper for recognizing faces in varied lighting circumstances. Significant steps in this suggested method include face detection using MTCNN, data normalization and pre-processing using random Gaussian noise, model creation for face classification, and authentication strategy. The model performance is tuned using the Monte Carlo dropout method. This step enhances the model's efficiency and increases face recognition accuracy.

References

[1] L. C. Jain, U. Halici, I. Hayashi, S. Lee, and S. Tsutsui, *Intelligent biometric techniques in fingerprint and face recognition*. CRC press, 1999.

[2] M. Chihaoui, A. Elkefi, W. Bellil, and C. Ben Amar, "A survey of 2D face recognition techniques," *Computerss,* vol. 5, no. 4, p. 21, 2016.

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

[4] S. Singh and S. Prasad, "Techniques and challenges of face recognition: A critical review," *Procedia computer science,* vol. 143, pp. 536-543, 2018.

[5] I. Adjabi, A. Ouahabi, A. Benzaoui, and A. Taleb-Ahmed, "Past, present, and future of face recognition: A review," *Electronics,* vol. 9, no. 8, p. 1188, 2020.

[6] 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.

[7] S. M. Pizer *et al.*, "Adaptive histogram equalization and its variations," *Computer vision, graphics, image processing,* vol. 39, no. 3, pp. 355-368, 1987.

[8] S. Shan, W. Gao, B. Cao, and D. Zhao, "Illumination normalization for robust face recognition against varying lighting conditions," in *2003 IEEE International SOI Conference. Proceedings (Cat. No. 03CH37443)*, 2003, pp. 157-164: IEEE.

[9] D. W. Jacobs, P. N. Belhumeur, and R. Basri, "Comparing images under variable illumination," in *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231)*, 1998, pp. 610-617: IEEE.

[10] J. Tumblin and G. Turk, "LCIS: A boundary hierarchy for detail-preserving contrast reduction," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 1999, pp. 83-90.

[11] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," 1994.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:.* 2014.

[13] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," 2015.

[14] J.-C. Chen, V. M. Patel, and R. Chellappa, "Unconstrained face verification using deep cnn features," in *2016 IEEE winter conference on applications of computer vision (WACV)*, 2016, pp. 1-9: IEEE.

[15] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," 2014.

[16] J.-S. Chan, G.-S. J. Hsu, H.-C. Shie, and Y.-X. Chen, "Face recognition by facial attribute assisted network," in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 3825-3829: IEEE.

[17] Z. An, W. Deng, T. Yuan, and J. Hu, "Deep transfer network with 3D morphable models for face recognition," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, 2018, pp. 416-422: IEEE.

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

[19] A. Kumar, A. Kaur, and M. Kumar, "Face detection techniques: a review," *Artificial Intelligence Review,* vol. 52, no. 2, pp. 927-948, 2019.

[20] D. T. P. Hapsari, C. G. Berliana, P. Winda, and M. A. Soeleman, "Face detection using Haar cascade in difference illumination," in *2018 International Seminar on Application for Technology of Information and Communication*, 2018, pp. 555-559: IEEE.

[21] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE signal processing letters,* vol. 23, no. 10, pp. 1499-1503, 2016.

[22] K. A. Majeed, "Face Detectors Evaluation to Select the Fastest among DLIB, HAAR Cascade, and MTCNN," 2021.

[23] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," 2017.

[24] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050-1059: PMLR.