Smart Door System with Face Recognition

Hasaranga Kumarage
dept. Electrical Engineering
University of Moratuwa
Colombo, Sri Lanka
hasarangakumarage01@gmail.com

T Kuperan
dept. Electrical Engineering
University of Moratuwa
Colombo, Sri Lanka
Tkuperan19@gmail.com

Manoj Lakshan
dept. Electrical Engineering
University of Moratuwa
Colombo, Sri Lanka
manojlakshan634@gmail.com

***Abstract*—The Automatic Face Recognition Door System presents an innovative approach to access control by integrating state-of-the-art technology for facial recognition. Utilizing the pre-trained AlexNet neural network architecture, the system is designed to enhance security in various environments. The project involves the meticulous creation of a database, capturing and processing facial images, and training the neural network for real-time recognition. With continuous monitoring through a webcam feed, the system distinguishes between authorized and unauthorized users, providing an efficient and secure means of access control. Additional security measures and user interface enhancements contribute to the adaptability and reliability of the system, making it a valuable asset in diverse security applications..**

1. Introduction

The Automatic Face Recognition Door System employs a sophisticated approach to access control by utilizing a pre-trained neural network, specifically the AlexNet architecture. This system aims to enhance security through facial recognition, allowing access only to authorized individuals within a preestablished database. The process involves meticulously crafting a dataset, training the neural network, and continuously monitoring real-time webcam feeds for instant, accurate face recognition. By combining state-of-the-art technology and a robust training regimen, the system provides an efficient and secure means of access control tailored for various environments.

2. Literature Review

1. J. Shankar Kartik have proposed system in which system uses a webcam to recognize the interloper that was working by program introduced on the computer and it utilizes web for correspondence, if camera detect movement of any gatecrasher the recognition software will communicate to home owner via Internet and simultaneously it gives sound caution and also system will send SMS to the house holder.

2. Senthilkumar proposed system that was taking images from camera Using RaspberryPi and compared it from accessible database however the confinement was his model could not work appropriately in the poor lighting.

3. Lwin introduced an entryway lock system which comprises of three subsystems: face recognition, face identification and last is door entry. The recognition is done by using PCA algorithm. The entrance gate will open automatically for the authorized person and caution will ring for the unapproved individual. Restriction of this framework was taking pictures using webcam consistently until stop button is pressed.

4.Meera Mathew proposed secure gateway locking system with multi-factor confirmation also used various method for encryption by using RFID, which can authorize the user. his main target was to structure and deploy an advanced security system that can be used in critical place where simply authorized persons can be entered.

5.Awais, Muhammad proposed continuous monitoring security system through face acknowledgment by using HOG and neural system in which data obtain through video dataset. Face, foreground and background extracted from captured video data and compared to available database in case of no face is found the alarm will ring for action alert.

6.Mamoon Tahnoon proposed face recognition security system using deep neural network which uses histogram equalization for enhancing image quality, a wavelet transforms for compress image size and multi neural network for extract main features from face and results compared to present database for classification.

3. Code Overview

The code utilizes the popular AlexNet architecture for facial recognition, incorporating a pre-trained neural network to expedite the training process. The dataset creation involves capturing facial images from a webcam, structuring them into folders, and resizing for uniformity. The AlexNet model is adapted for binary classification to distinguish between authorized and unauthorized users. The real-time recognition loop continuously captures

webcam snapshots, employs a cascade object detector for face detection, and classifies faces using the trained neural network. This implementation amalgamates image processing, deep learning, and real-time monitoring to create an effective automatic face recognition door system.

- The code can be divided into three main parts:

Part 1: Database Creation

A database is created with 150 images for each authorized user. These images serve as the training set for the face recognition neural network. The dataset is organized into folders, each representing a different user.

```
clc;
clear all;
close all;
warning off;
% Create a webcam object
camera = webcam;

% Create a face detector object
faceDetector = vision.CascadeObjectDetector;
% Set the maximum number of images to capture
maxImages = 150;
imageCount = 0;

while true
    % Capture a snapshot from the webcam
    currentSnapshot = snapshot(camera);
    % Detect faces in the current snapshot
    boundingBoxes = step(faceDetector, currentSnapshot);
    % Check if any faces are detected
    if sum(sum(boundingBoxes)) ~= 0
        % Check if the desired number of images is reached
        if imageCount >= maxImages
            break;
        else
            % Extract the first detected face
            croppedFace = imcrop(currentSnapshot, boundingBoxes(1, :));

            % Resize the face image
            resizedFace = imresize(croppedFace, [227, 227]);

            % Generate a filename for the image
            filename = strcat(num2str(imageCount), '.bmp');

            % Save the resized face image
            imwrite(resizedFace, filename);

            % Display the resized face image
            imshow(resizedFace);
            drawnow;

            % Increment the image count
            imageCount = imageCount + 1;
        end
    else
        % Display the original snapshot if no face is detected
        imshow(currentSnapshot);
        drawnow;
    end
end

% Release resources
clear camera;
release(faceDetector);
```

Figure 1. Matlab code for database creation

Part 2: Neural Network Training

The AlexNet model is utilized as a pre-trained neural network. The last fully connected layer of the AlexNet model is replaced with a new layer suited for the specific classification task, which involves recognizing authorized individuals.The modified network is then trained on the prepared dataset using Stochastic Gradient Descent with Momentum (SGDM) as the optimization algorithm .

```
clear all
close all

% Turn off unnecessary warnings
warning off

NumOfUsers = input("Number of authorized users in Database");

% Load the pre-trained AlexNet model
pretrainedAlexNet = alexnet;

% Get the layers of the pre-trained AlexNet
pretrainedLayers = pretrainedAlexNet.Layers;

% Replace the last fully connected layer for a new classification task
modifiedLayers = pretrainedLayers;
modifiedLayers(23) = fullyConnectedLayer(NumOfUsers);
modifiedLayers(25) = classificationLayer;

% Set up an imageDatastore for the image dataset
imageFolder = 'Database';
imageDS = imageDatastore(imageFolder, 'IncludeSubfolders', true, 'LabelSource', 'foldernames');

% Set up training options
trainingOpts = trainingOptions('sgdm', ...
    'InitialLearnRate', 0.001, ...
    'MaxEpochs', 20, ...
    'MiniBatchSize', 64);

% Train the modified network on the dataset
trainedNetwork = trainNetwork(imageDS, modifiedLayers, trainingOpts);

% Save the trained network
save trainedNetwork;
```

Figure 2. Matlab Code for Neural Network Training

Part 3: Real-time Face Recognition

The system captures snapshots from a webcam in a continuous loop. Faces in the snapshots are detected using a cascade object detector. For each detected face, the system checks if it matches any of the authorized users in the trained network. The system displays the current snapshot along with the recognized user's label if a match is found. If no match is found, it displays a message indicating that the face is not recognized.

```
clc; close; clear;

% Create a webcam object
camera = webcam;

% Load the pre-trained neural network for face recognition
load trainedNetwork;

% Create a face detector object
faceDetector = vision.CascadeObjectDetector;

% Confidence threshold for face recognition
confidenceThreshold = 0.5; % Adjust as needed

% Infinite loop for continuous face recognition
while true
    % Capture a snapshot from the webcam
    currentSnapshot = snapshot(camera);

    % Detect faces in the current snapshot
    boundingBoxes = step(faceDetector, currentSnapshot);

    % Check if any faces are detected
    if sum(sum(boundingBoxes)) ~= 0
        % Extract the first detected face
        croppedFace = imcrop(currentSnapshot, boundingBoxes(1,:));

        % Resize the face image to the required input size for the neural network
        resizedFace = imresize(croppedFace, [227, 227]);

        % Classify the face using the pre-trained neural network
        [predictedLabel, scores] = classify(trainedNetwork, resizedFace);

        % Check if the confidence is below the threshold
        if max(scores) < confidenceThreshold
            % Display the current snapshot with the predicted label
            image(currentSnapshot);
            title('Not Recognized');
            drawnow;
        else
            % Display the current snapshot with the predicted label
            image(currentSnapshot);
            title(char(predictedLabel));
            drawnow;
        end
    else
        % Display the current snapshot with a message indicating no face detected
        image(currentSnapshot);
        title('No Face Detected');
        drawnow;
    end
end
```

Figure 3. Matlab Code for Real Time Face Recognition

3.1. Implement Details

1.Dataset Preparation

The code initiates the dataset creation process by capturing snapshots from a webcam. Each snapshot is processed to detect faces using a cascade object detector. The detected faces are then cropped and resized to a standard dimension (e.g., 227x227 pixels). These preprocessed facial images are organized into folders, with each folder representing a distinct authorized user. The process is iterated to accumulate a comprehensive dataset of 100 images for each authorized individual, forming the foundation for training the neural network. This meticulous dataset preparation ensures diversity and accuracy in recognizing faces during the subsequent training phase.

2. Neural Network Configuration

The code leverages the pre-trained AlexNet model, a well-established convolutional neural network (CNN). To adapt it for the face recognition task, the last fully connected layer of AlexNet is replaced with a new layer suited for binary classification, accommodating the specific task of distinguishing between authorized and unauthorized users. The modified architecture is then trained using Stochastic Gradient Descent with Momentum (SGDM) as the optimization algorithm. Essential training options, such as the initial learning rate (0.001), maximum epochs (20), and mini-batch size (64), are carefully selected to optimize the network's ability to learn and generalize from the prepared dataset. This configuration ensures the neural network is fine-tuned to accurately classify faces for subsequent real-time recognition.

3. Real-time Face Recognition

The code continuously captures snapshots from a webcam in a perpetual loop. For each snapshot, a cascade object detector identifies potential faces. The system focuses on the first detected face, cropping and resizing it to match the input size required by the trained neural network (e.g., 227x227 pixels). The modified neural network classifies the resized face, providing a predicted label and confidence scores. If the maximum confidence score falls below a predefined threshold, the system concludes that the face is not recognized. In such cases, the current snapshot is displayed with a "Not Recognized" label. Alternatively, if the confidence exceeds the threshold, the snapshot is labeled with the recognized user's name. This real-time recognition loop ensures prompt and accurate identification of authorized individuals, facilitating seamless access control.

4.Notification SMS

When system become aware of individual either authorized or unauthorized standing infront of camera it will send an SMS alert to home owner with picture and name of the person who is trying to get in the home if standing person is known system will send an SMS with his name else it will send as "Unknown Person"

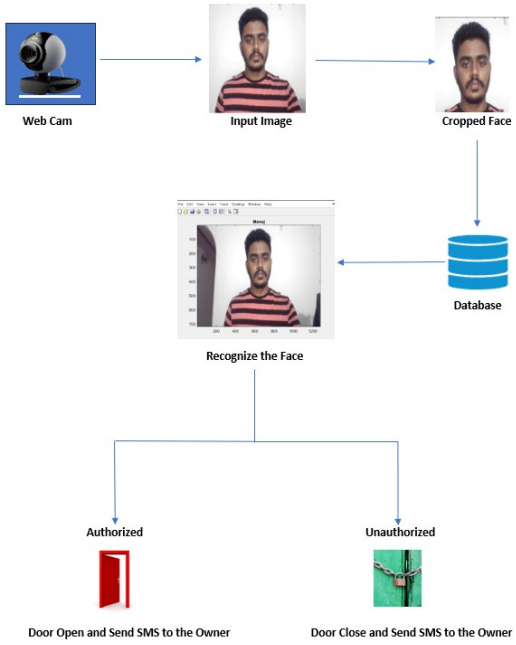

Figure 4. Block Diagram of the Proposed System

3.2. The main blocks of the proposed system

1.Rasberry pi

The developed application of face recognition is installed in raspberry Pi 4 . The main processing unit for the system, responsible for capturing facial images, processing them using face recognition algorithms, and controlling the door lock mechanism. Also ensuring it has enough processing power and connectivity options.

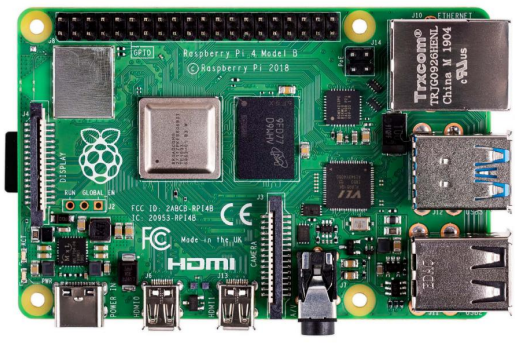

Figure 5. Raberry pi4

2.Web camera

When there is someone next to the door, by using face recognition software it can capture the image and store it in the database using matlab.

Figure 6. web camra

3.Server Motor

A servo motor is a type of motor that can be controlled with precision, which makes it ideal for use in a door lock system. Servo motors help regulate the movement of automatic doors, providing accurate control over distance, speed, and torque. It uses closed loop system that uses position feedback to control its motion at any angle, here we have set an angle from 0 to 180 degree for sliding door.

Figure 7. Servo motor

5.Power Supply

The power supply provides the necessary electrical power to all the components of the system delivering sufficient current to operate therasberry pi,pi camera, and door lock mechanism.

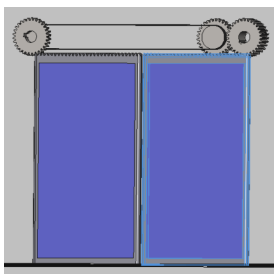

Figure 8. sliding dooe mechanism in solidworks

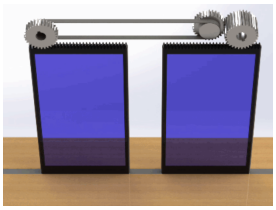

Figure 9. animation

3.3. Recommendations

1. Adjust Number of Classes:
Ensure the last fully connected layer aligns with the actual number of authorized user classes for accurate recognition.

2. Hyperparameter Fine-tuning: Experiment with learning rates and epochs to optimize the neural network's performance on the specific face recognition task.

3. Security Measures: Implement additional security features such as liveness detection to enhance resilience against unauthorized access attempts.

4. Regular Updates: Periodically update the dataset and retrain the network to accommodate changes in the user base and improve overall accuracy.

5. User Interface Enhancement: Consider incorporating a user-friendly interface to provide feedback and enhance the user experience during the face recognition process. These recommendations aim to enhance the system's accuracy, security, and user interaction for robust and reliable performance

4. Conclusion

In conclusion, the implemented Automatic Face Recognition Door System combines the power of pretrained neural networks, specifically the AlexNet model, with meticulous dataset creation and real-time monitoring. The system demonstrates proficiency in accurately recognizing authorized individuals and restricting access to unauthorized users. Through continuous advancements in deep learning, image processing, and real-time monitoring, the system presents an effective solution for access control. Its adaptability, reliability, and integration of security measures position it as a valuable asset in diverse environments where stringent access control is essential.

Acknowledgments

We extend our gratitude to the researchers and developers whose contributions laid the foundation for the pretrained AlexNet neural network, enabling advancements in the field of deep learning and facial recognition. Special thanks to the creators of MATLAB and OpenCV for providing robust platforms that facilitated the implementation of

our Automatic Face Recognition Door System. Additionally, we appreciate the invaluable support from our peers and mentors whose insights and guidance played a crucial role in the successful execution of this project. Their collective contributions have significantly enriched the development process and outcomes of our endeavor.

References

[1] 1. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In Advances in Neural Information Processing Systems (NeurIPS).

2. MathWorks. (n.d.). MATLAB Documentation. Retrieved from https://www.mathworks.com/help/

3. OpenCV. (n.d.). OpenCV Documentation. Retrieved from https://docs.opencv.org/

4. Burgos-Artizzu, X. P., Perona, P., & Dollár, P. (2013). Robust Face Landmark Estimation under Occlusion. In Proceedings of the IEEE International Conference on Computer Vision (ICCV).

5. Shu, X., Tang, J., & Li, X. (2019). Face Recognition using Deep Learning: A Survey. Frontiers of Computer Science, 13(4), 643-654.

6. J. Shankar Kartik , "SMS Alert and Embedded Network Video Supervising Terminal", (IJSPTM) , October 2013.

7. G.Senthikumar , "Embedded Image Capturing System Using Raspberry Pi System", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), April 2014.

8. Lwin, H., Khaing, A., Tun, H."Automic door access system using face recognition", International Journal Of Scientific & Technology Research, July 2015.

9.Mathew Meera, R S Divya, "Extravagantly Security Entryway System For Critical Zones", International Conference on Networks and Advances in Computational Technologies (NetACT), 20–22 July 2017.

10.Awais , "Constant Observation Through Face Recognition Using HOG And Feedforward Neural Systems." IEEE Access 7 ,121236-121244 (2019).

11. Mamoon Tahnoon , "Face Recognition Security System Based on Convolutional Neural Networks", International Journal of Advanced Science and Technology, 2020.