IoT based Human Detection System with Neural Network Classifier

Armanur Rahman

ID: 2013-3-60-034

Md. Bappi Parvez

ID: 2013-3-60-017

A thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering.

Department of Computer Science and Engineering
East West University
Dhaka-1212, Bangladesh

August, 2017

Declaration

This thesis has been submitted to the department of Computer Science and Engineering, East West University in the partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science and Engineering by us under the supervision of Dr. Ahmed Wasif Reza, Associate Professor at Department of CSE at East West University under the course 'CSE 497'. We also declare that this thesis has not been submitted elsewhere for the requirement of any degree or any other purposes. This thesis complies with the regulations of this University and meets the accepted standards with respect to originality and quality. We hereby release this thesis to the public. We also authorize the University or other individuals to make copies of this thesis as needed for scholarly research.

Armanur Rahman

ID: 2013-3-60-034

Department of Computer Science and Engineering

East West University.

Md. Bappi Parvez

ID: 2013-3-60-017

Department of Computer Science and Engineering

East West University.

Letter of Acceptance

The thesis entitled “IoT based human detection system with Neural Network Classifier” submitted by Armanur Rahman, ID 2013-3-60-034 & Md. Bappi Parvez, ID 2013-3-60-017 to the department of Computer Science & Engineering, East West University, Dhaka 1212, Bangladesh is accepted as satisfactory for partial fulfillments for the degree of Bachelor of Science in Computer Science & Engineering in August 2017.

Board of Examiners

1________________________________

Dr. Ahmed Wasif Reza

Associate Professor Supervisor

Department of Computer Science and Engineering

East West University, Dhaka, Bangladesh

2________________________________

Dr. Md. MozammelHuq Azad Khan

Professor and Chairperson Chairperson

Department of Computer Science and Engineering

East West University, Dhaka, Bangladesh

Acknowledgements

First of all, we are grateful to the Almighty God for establishing me to complete this research. Therefore, we would not like to make efforts to find best words to express my thankfulness other than simply listing those people who have contributed to this thesis itself in an essentialway. We wish to express my sincere thanks and gratitude to my advisor Dr. Ahmed Wasif Reza, Associate Professor at Dept. of CSE for the continuous support during my thesis study and related research, for his patience, motivation, and immense knowledge. His guidance helped us in all the time of research and writing of the thesis. We will always be grateful for having the opportunity to study under him.

We are thankful to all of my teachers, Department of CSE, East West University. We also grateful to all of my college, primary and secondary school teachers who are my first teachers in my life and initiator of my basic knowledge.

We would like to express our thanks to our parents and siblings for supporting us spiritually throughout writing this thesis. And we are thankful to all my friends and colleagues. And at last we again thanks to the creator Allah for everything.

Abstract

Neural network is a programming paradigm which enable a computer to learn from observational data. Neural network was created based on the human neuron. Now a day's Neural network provide more accurate solution in the field of computer vision such as image recognition, face detection and many other field speech recognition, natural language processing. To make model using neural network, one of the hardest part is collect training data. To train a model a neural network need a huge amount of data. In this study, we introduce a technique how to customize training data so that using same amount of data we can achieve better accuracy. As a neural network model, we have used convolutional neural network. Our problem is human detection. We give an input image in our model our model can predict the input image is human or not. We have collected human face image as positive image and images those not contain human face as negative image. Then we build a convolutional neural network which can predict human or non-human. We tune our training data in 2 processes then again train our model which can achieve more accuracy than the previous one. We took 3 type of test datasets and compare those test set result.Finally, we implement our model for IoT based smart door system.

Contents

Contents		Page
List of Figure		**Vii**
List of Tables		**ix**
List of Abbreviations		**X**
1	**Introduction**	**01-03**
1.1	Background	01
1.2	Problem Statements	02
1.3	Research objective	02
1.4	Thesis contribution	02
1.5	Paper Orientation	03
2	**Literature Review**	**04-05**
2.1	Existing Technique	04-05
2.2	Summery	05
3	**Research methodology**	**06-19**
3.1	Proposed Framework	06-08
3.2	Proposed Network Architecture	09-12
3.3	Proposed Algorithm	13-17
3.4	Proposed System	18-19

3.5	Summary	19
4	**Results and Discussion**	**20-37**
4.1	Experimental Scenario	20-34
4.2	Performance Comparison	35-36
4.3	Summary	37
5	**Conclusion**	**38**
5.1	Overall Conclusion	38
5.2	Future works	38
	References	**39-40**

List of Figures

Figures		Page
3.1	Dataflow diagram of model	07
3.2	Diagram of CNN model	10
3.3	Original image and flip image example	14
3.4	Original image and rotate image example	15
3.5	Diagram of proposed system	19
4.1	Pie chart of raw data model for training dataset 1	21
4.2	Pie chart of modified data model for training dataset 1	23
4.3	Accuracy graph for test dataset 1	24
4.4	Pie chart of raw data model for training dataset 2	26
4.5	Pie chart of modified data model for training dataset 2	28
4.6	Accuracy graph for test dataset 2	29
4.7	Pie chart of raw data model for training dataset 3	31
4.8	Pie chart of modified data model for training dataset 3	33
4.9	Accuracy graph for test dataset 3	34
4.10	Comparison of three test dataset	36

List of Tables

Tables		Page
3.1	Confusion Matrix.	16
4.1	Confusion matrix of raw data model for Test Dataset1	20
4.2	Details Accuracy of raw data model for test dataset 1	21
4.3	Confusion matrix of Modified data model for Test Dataset1	22
4.4	Details Accuracy of modified data model for test dataset 1	23
4.5	Confusion matrix of raw data model for Test Dataset2	25
4.6	Details Accuracy of raw data model for test dataset 2	26
4.7	Confusion matrix of modified data model for Test Dataset2	27
4.8	Details Accuracy of modified data model for test dataset 2	27
4.9	Confusion matrix of raw data model for Test Dataset3	30
4.10	Details Accuracy of raw data model for test dataset3	31
4.11	Confusion matrix of modified data model for Test Dataset3	32
4.12	Details Accuracy of modified data model for test dataset3	33
4.13	Comparison amount three dataset	35

List of Abbreviation

CNN	Convolutional neural network
IoT	Internet of things
SVM	Support Vector Machine
Relu	Rectified Linear unit
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
CV	Computer Vision
FCNN	Fully convolutional neural network

Chapter 1

Introduction

1.1 Background

Human visual system is one of the wonders of the world. They are skilled at adopt various kinds of human face detection. But when same things come for computer, it’s not easy to detect face from image. There are more than billions of species in this world. Among them detection human face is little tough. Besides an image does not contain only face, maximum area of the image does not contain faces. Using the technique of data mining and deep learning algorithm predicts an image contain human face. Deep learning, in particular convolutional neural network (CNN), has achieved promising result in face detection recently. In real world face detection large visual variations, such as those due to pose, expression and lighting effect. CNN can take image as direct input and powerful to rotation, translation and scaling deformation of images. By manually acquiring some facial feature from an image is difficult and time consuming where CNN could extract effective facial feature automatically. So, choosing CNN for face detection is benefited. Thus, object detection has become one of the most important in computer vision field, especially IoT devices like surveillance and biometrics device. In this research, we built an advanced discriminative model to accurately detect faces from an image weather input image does contain human face or not. The target of this paper is to propose a IoT based smart door system with CNN where if the device found human face only then door will open.

1.2 Problem Statements.

i. From the problem of getting better accuracy neural network needs more training data, we introduce our model where with the modification of existing dataset we increase the accuracy. We also propose a generalize framework that will work for any image object detection.

ii. From the object detection based door open system, we proposed an IoT based smart door system architecture with our model.

1.3 Research objective

i. To propose an algorithm which will give better accuracy with same data set.

ii. To design IoT based smart door system with CNN classifier.

1.4 Thesis contribution

i. We have introduced an algorithm which increase accuracy with same amount of data previously used. Our algorithm maximizes the model accuracy without increase the amount of training data. We showed the difference of accuracy between raw data and modified data.

ii. Then we have proposed an IoT based smart door system where we integrate our introduced technique. In which, door will open only when it found human face in front of door. Otherwise door will keep close.

1.5 Paper Orientation

i. In chapter one, we have discussed the background of our research, the objective of our research and our contribution in this study.

ii. In chapter two, we have discussed the literature review, paper and resource from where we inspire.

iii. In chapter three, we show our proposed model. Proposed method framework, proposed network architecture, proposed algorithm, Different mathematical equations and examples.

iv. In chapter four, we show results of different experimental senate, and make a comparison of those result.

v. In chapter five, we discuss the overall conclusion, and also discuss about our future work.

Chapter 2

Literature Review

2.1 Review of Existing Techniques

There are many kinds of research studies have been done using artificial intelligence, machine learning and deep learning in computer vision sector. The classical object detection algorithms always use the sliding window method where image detected with a fixed-size rectangular window. Then the classifier will decide on that window. There are various classifiers based on different decision method like feature-based techniques, deep learning neural networks [12] and statistical learning-based techniques [13]. In paper [4], author proposed a neural network and they apply CNN for face detection of an image. with their network, they scanned whole image at all possible locations. In reference [6], the author develops a neural network that can detect semi-frontal human faces in complex images. In paper [1], the author proposed a face recognition system based on CNN and SVM. In their system, they used CNN to extract image feature and SVM as a classifier. They used optimization technique to improve performance of CNN. They used some necessary data to pre-train CNN to improve the generalization ability of that network. The internet of things (IoT) is an emerging topic of technical, social and economic significance. In paper [10], The author has introduced an heterogenous many core based object detection system for embedded system and IoT device. Their face detection showed the implementation of the framework. In reference [11], They introduced, wireless access monitoring and control system based on digital door lock. They used ZigBee human detection module to identify the human using camera. After identification, the digital door will open which is operates a motor connected locking system. In paper [14], They introduced an approach salient object detection mainly it combination of multi-task fully convolutional neural network(FCNN) and nonlinear regression. For an image, they first use FCNN to compute a course saliency map as a foreground information and then use non-linear regression for saliency refinement. After that, the course-grained foreground and background saliency map are combined and finally refined based on graph Laplacian regularized non-linear regression to identify actual information of object boundary.

2.2 Summary

Early approaches in this field typically formulate new algorithm or new IoT based technique for face detection as the problem of image processing. To improve model accuracy in neural network, increase the training set data. But we introduced a model where without increasing training data we can improve the accuracy of a model. We also propose an IoT based door lock system which is implementation of our model.

Chapter 3

Research methodology

3.1 Proposed Framework

In this study, we have used 22200 images as positive image which contain human and 15815 images as negative images which does not any human images. We have collect positive images from Face Recognition Database [3], LFWcrop [9], web face [8] and negative images from visual geometry group [5], Ponce Research Group[7].

To train a convolution neural network, we have to use a large number of training data. But collecting those training data is not easy, so we make 2 steps pre- processing our training dataset so that we can achieve more accuracy from the same amount of dataset. In the first pre-processing, we flip our 14% positive training dataset from left to right. In the second pre-processing, we rotate our 14% positive training images rotate from 0-25 degree randomly. The reason of rotate and flip our 28% positive training dataset is that our model can extract more feature from the training data. In real world, human face can appear in different angle, from different side. So, if we flip and rotate some portion of our training dataset we will get more accurate result.

In our proposed framework, we show how we modify our training dataset with two processes and how the data are flow to the CNN model.

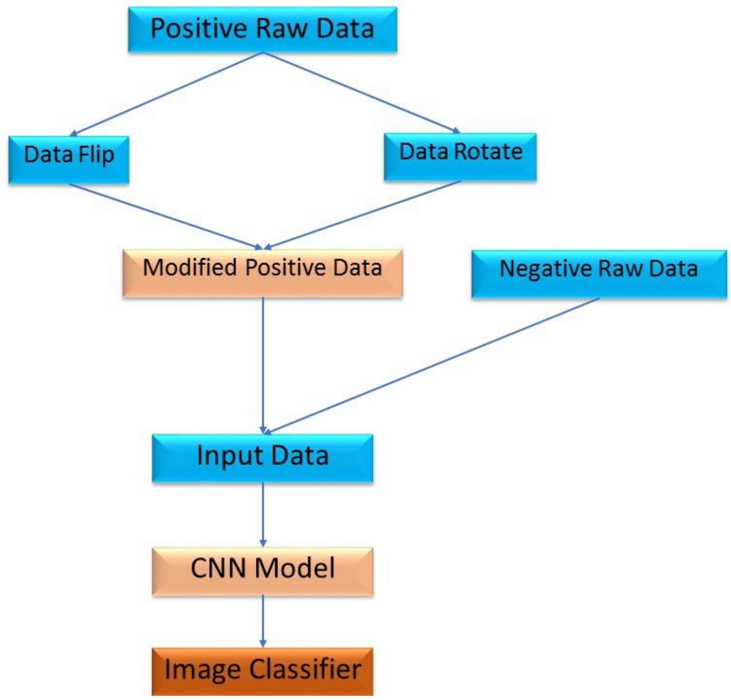


Figure 3.1 Dataflow Diagram of Model

Figure 3.1 shows how our algorithm transforms training data to modified data to get better accuracy and without increasing the amount of data for training data set.

Positive raw data: Images that collect from different source that contains only human images.

Flip data: we flip 14% of positive raw data flip from left to right

Rotate data: 14% of our positive raw data are being rotated from left to right in different angle. The angle of the rotation has been selected randomly.

Modified data: 28% of raw and flip data arrange together with other 78% of positive raw data.

Raw negative data: Data that does not contain any human images collected from different source.

Input data: modified positive data and negative data has now become the input data. Which we can use to train our convolutional neural network.

3.2 Proposed Network Architecture

We have created a Convolutional neural network using 3 convolution layers 2 max pooling layer and 2 fully connected layers and a dropout. Input image with size 32*32 and channel one is inserted into the first convolution layer.

The layer is as sequence,

- First convolution layer
- First max pooling layer
- Second convolution layer
- Third convolution layer
- Second max pooling layer
- First fully connected layer
- Dropout
- Second fully connected layer

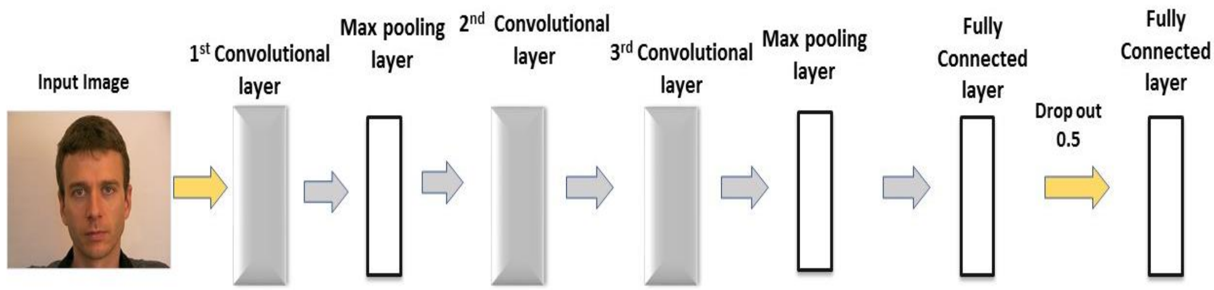


Figure 3.2: Diagram of CNN model

Figure 3.2 Shows how we design our network. It also shows the combination of using 3 convolution layers 2 max pooling layer and 2 fully connected layers and a dropout of Convolutional neural network.

1. First Convolution Layer

In first convolution, we have used same padding strive [1,1,1,1]. Input images with height 32 weight 32. Number of filter 32 filter size 3 Relu has been used as activation function. Bias are adding to weight randomly.

2. First max pooling

First convolution layer output will be used as the output of the First max pooling layer. Size of the kernel or pooling kernel size as 2, padding also used as same.

3. Second Convolution Layer

First max pooling layer output has been used as the input of the Second Convolution layer. In Second convolution, we have used same padding strive [1,1,1,1]. Number of filter 64 filter size 3 Relu has been used as activation function. Bias are added to weight randomly.

4. Third Convolution Layer

Second convolution layer output has been used as the input of the Third Convolution layer. In Third convolution, we have used same padding strive [1,1,1,1]. Number of filter 64 filter size 3 Relu has been used as activation function. Bias are added to weight randomly.

5. Second max pooling

Third convolution layer output will be used as the output of the Second max pooling layer. Size of the kernel or pooling kernel size as 2, padding also used as same.

6. First fully connected layer

Second max pooling output has been used as the input of the First fully connected layer. Number of units for this layer is 512. Relu is the activation function of this layer.

7. Dropout

We have used 0.5 dropout rate, it will throw randomly some date to prevent over fit.

8. Second fully connected layer

The units of this fully connected layer is Second, because in this layer it will predict the final result. SoftMax has been used as the activation function.

3.3 Proposed Algorithm

Step1: dataset = read all positive image data

Step2: unflipdata = 14% of dataset

Step3: while (unfilpdata)

    Step4: flipdata = flip data from left to right

    Step5: save flip data in specific destination

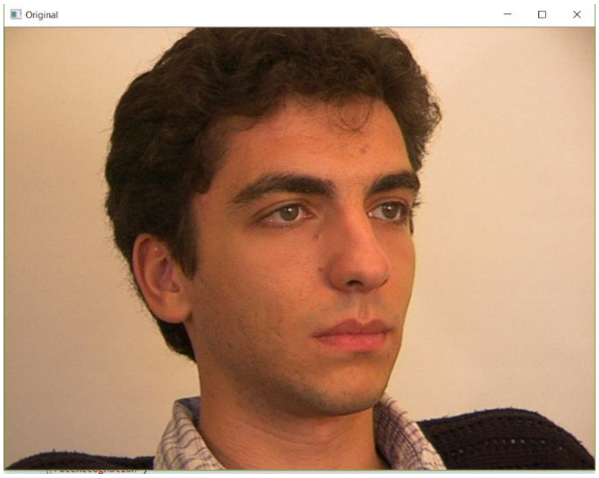
Step6: unrotatedata = 14% of dataset

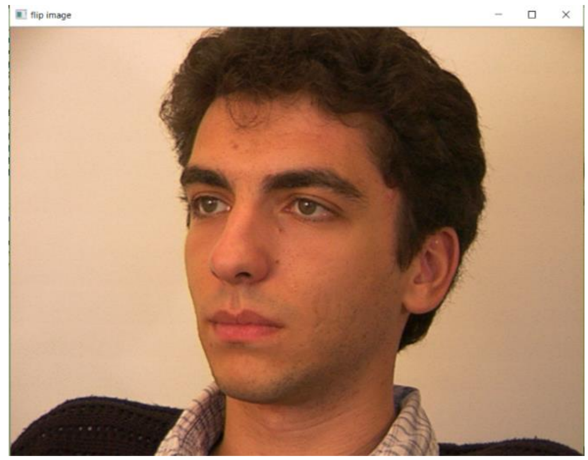
Step7: while(unrotatedata)

    Step8: rotatedata = rotate data from 0-30 degree randomly

    Step9: save rotate data in the specific destination

Step10: modifiedimputdata = move flipdata and rotatedata to positive image data

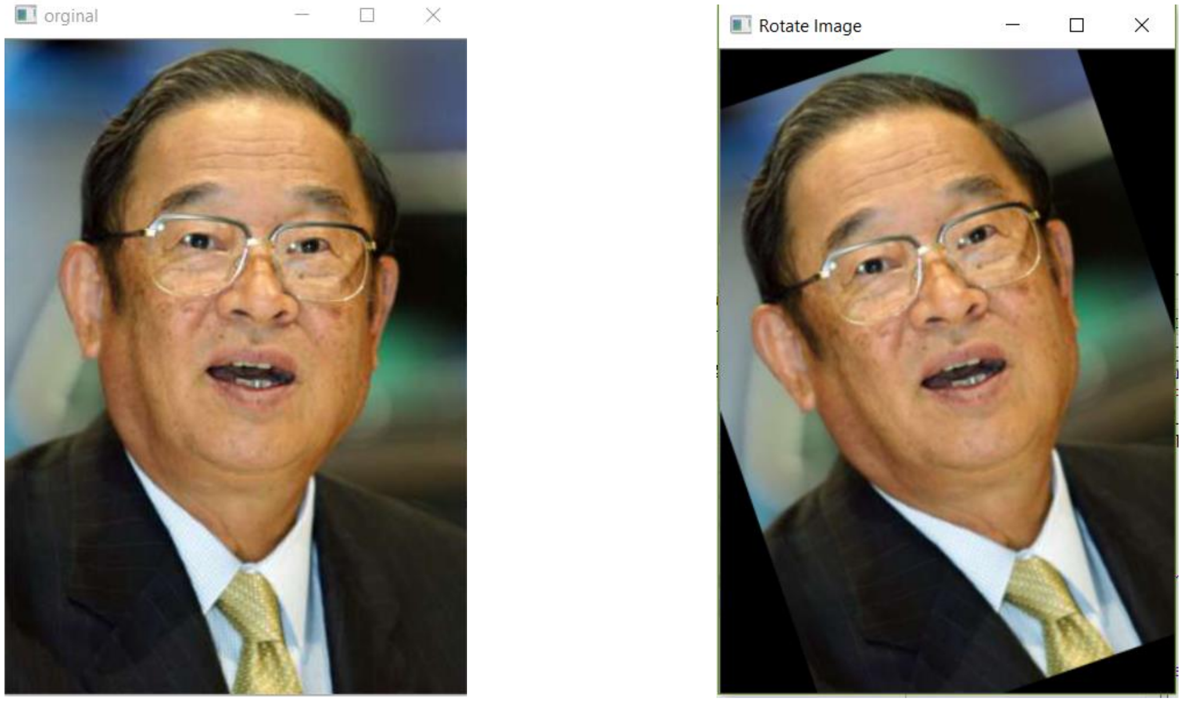




Orginal image flip image

Figure 3.3 Original image and flip image example

In figure 3.3 first image show one of the raw data images after passing the raw image the second image shows the modified image which is flip from left to right.



Orginal Image

Rotate Image

Figure 3.4 Orginal image and Rotate image Example

In figure 3.4 first image show one of the raw data images after passing the raw image in our algorithm the second image shows the modified image which is rotate with a random degree.

Mathematics Equation

Confusion matrixes mainly describe the performance of a classifier model and it contains information about actual and predicted classifications done by a classification system [24]. For evaluating the performance of such systems, we have to use the data in the matrix. The following table shows the confusion matrix for a two-class classifier.

- **TN** is the number of **correct** predictions that an instance is **negative**.
- **FP** is the number of **incorrect** predictions that an instance is **positive**.
- **FN** is the number of **incorrect** of predictions that an instance **negative**.
- **TP** is the number of **correct** predictions that an instance is **positive**.

Table 3.1: Confusion Matrix.

		Predicted	
		positive	Negative
Actual	Positive	**TP**	**FN**
	Negative	**FP**	**TN**

We need a couple of mathematical formula to calculate the accuracy, precision, recall and F1 measure. With the help of below formulas, we calculate accuracy and other staffs and those are showing in the result section.

i) For decision tree: Calculating the information gain

$$\text{True positive rate} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(3.1)}$$

$$\text{False positive rate} = \frac{\boldsymbol{FP}}{\boldsymbol{FP+TN}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(3.2)}$$

$$\text{Precision (confidence)} = \frac{TP}{TP+FP} \ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(3.3)}$$

$$\text{Recall} = \frac{TP}{TP+FN} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(3.4)}$$

$$\text{F1 measure} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \ldots\ldots\ldots\ldots\ldots\ldots \text{(3.5)}$$

ii) Evaluation the Performance: For overall accuracy,

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+FP+TN+FN)} \times 100\% \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(3.6)}$$

3.4 Proposed System

In this propose system there will a web camera with every door. When any human stand in front of the door the door the camera captures the image, and send to the server with its door id, our human classifier model will run in the server it checks the input image whether it contains human or not, if the classifier detects human it sends signal to the microcontroller of the specific door from which door the input image come from. Then the door will open automatically.

Proposed System Diagram

In our proposed system, there will be individual web camera attach each door and every door has a unique door id, so that server can send signal to each individual door. The web camera capture images and send it to the server, the server will make decision whether it opens the door or not. If the server detects any human in the input image it will sent signal to the microcontroller of that specific door to open the door otherwise server takes no action. The human detection part is performed by our CNN classifier .

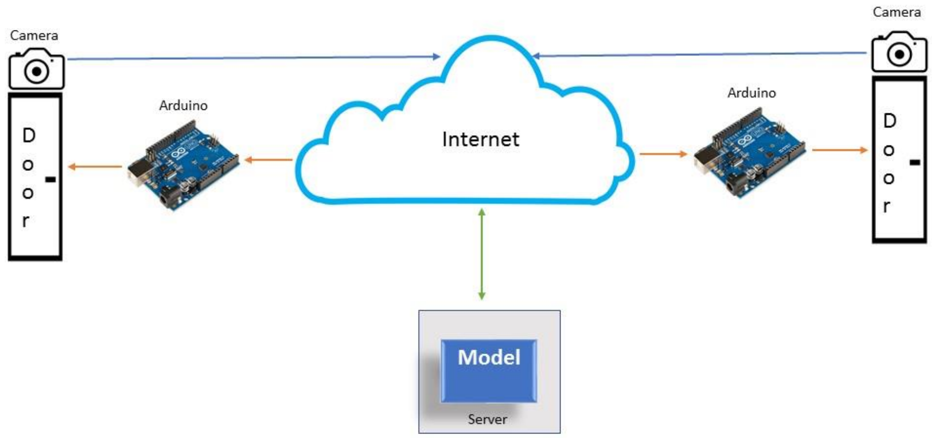


Figure 3.5: Diagram of proposed system

Figure 3.5 shows our proposed system. In this system, there will be a web camera with every door and the door will be identified with a unique door id. The camera will send capture image with its door id to the server. Our human classifier will run in the server and check whether the input image contain any human. If the classifier found human the server send signal to the micro controller of the specific door number to open the door.

3.5 Summary

From this chapter, we get a CNN model that can detect human and we also introduced a method so that we can modify our training dataset. We give some examples how our modified data look like. Then we propose an IoT based system where we can implement our model. In the next chapter, we compare the result of two types of dataset.

Chapter 4

Result and discussion

4.1 Experimental Scenario

In this study, first we have trained our convolution neural network with raw dataset and calculate the accuracy of the model. Then we train our CNN model with our modified dataset and calculate the accuracy. Then we compare our model for raw dataset and training dataset.

Test Dataset 1

Total tested data 1848. Human face contains data 1403 Non-human face 445

Confusion matrix for raw data model

Table 4.1 Confusion matrix of raw data model for Test Dataset1

		Predicted	
		Human	Non-human
Actual	Human	**956**	**447**
	Non-human	**90**	**355**

Table 4.1 the confusion matrix shows the value of TP, TN, FP, and FN.

Here,

TP = 956, TN = 355, FP = 90, FN = 447

Detail accuracy by class for raw data model

Table 4.2 Details Accuracy of raw data model for test dataset 1

Class	**value**
True positive rate	0.68
False positive rate	0.20
recall	0.91
Precision	0.68
F1 measure	0.78

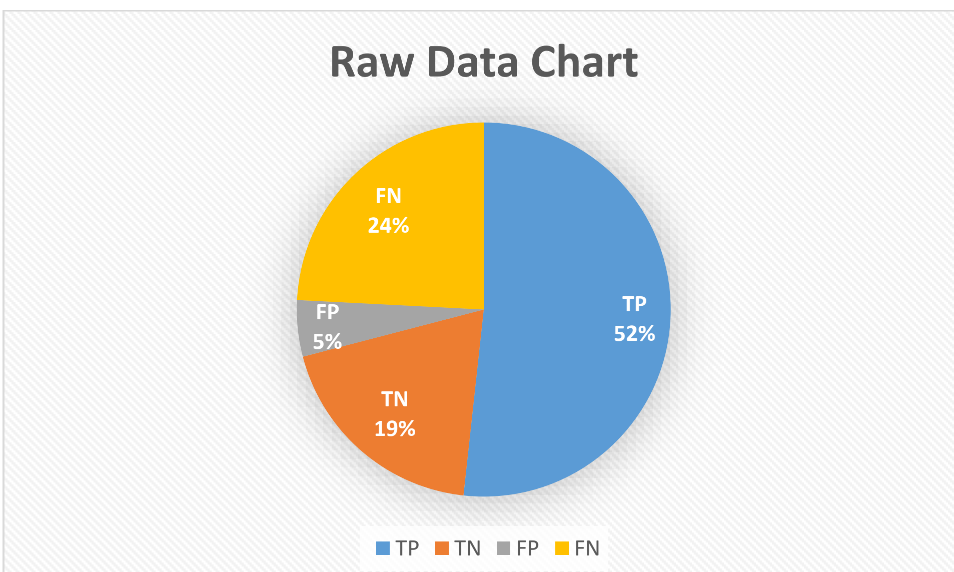


Figure 4.1 Pie chart of raw data model for training dataset 1

Figure 4.1 Shows that from raw data model we have achieved 52% TP and 19% TN in case of test dataset1. So, the raw model can successfully detect 71% test data and failed to detect 29% test data.

From Raw data, we have achieved 70.94% accuracy.

Confusion matrix for modified data,

Table 4.3 Confusion matrix of Modified data model for Test Dataset1

		Predicted	
		Human	Non-human
Actual	Human	1046	**357**
	Non-human	**104**	**341**

Table 4.3 the confusion matrix shows the value of TP, TN, FP, and FN.

Here,

TP = 1046, TN = 341, FP = 104, FN = 357

Detail accuracy by class for modified data model,

Table 4.4 Details Accuracy of modified data model for test dataset 1

Class	**value**
True positive rate	0.75
False positive rate	0.23
recall	0.91
Precision	0.75
F1 measure	0.82

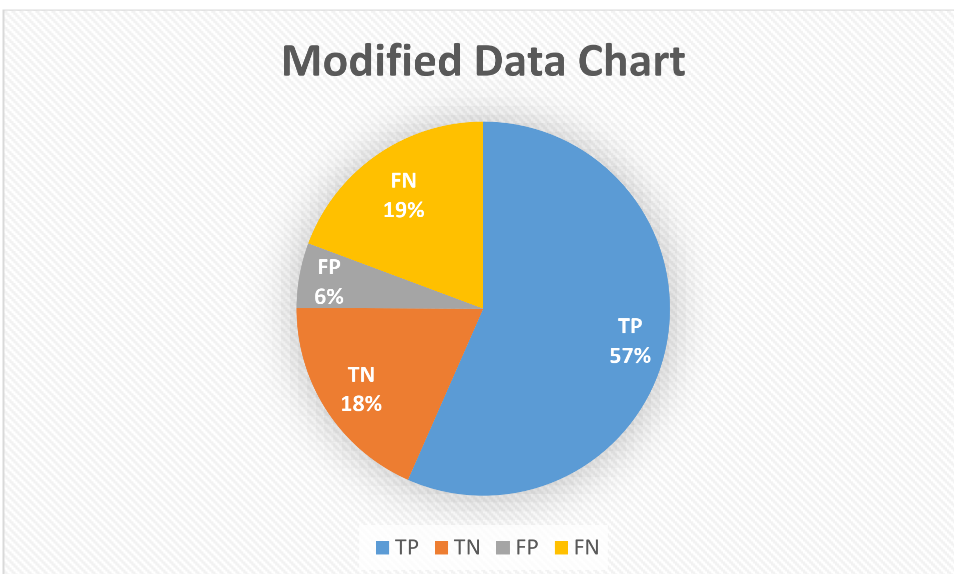


Figure 4.2 Pie chart of modified data model for test dataset 1

Figure 4.2 Shows that from raw data model we have achieved 57% TP and 18% TN in case of test dataset1. So, the modified data model can successfully detect 75% test data and failed to detect 25% test data.

From modified data, we have achieved 75.05% accuracy

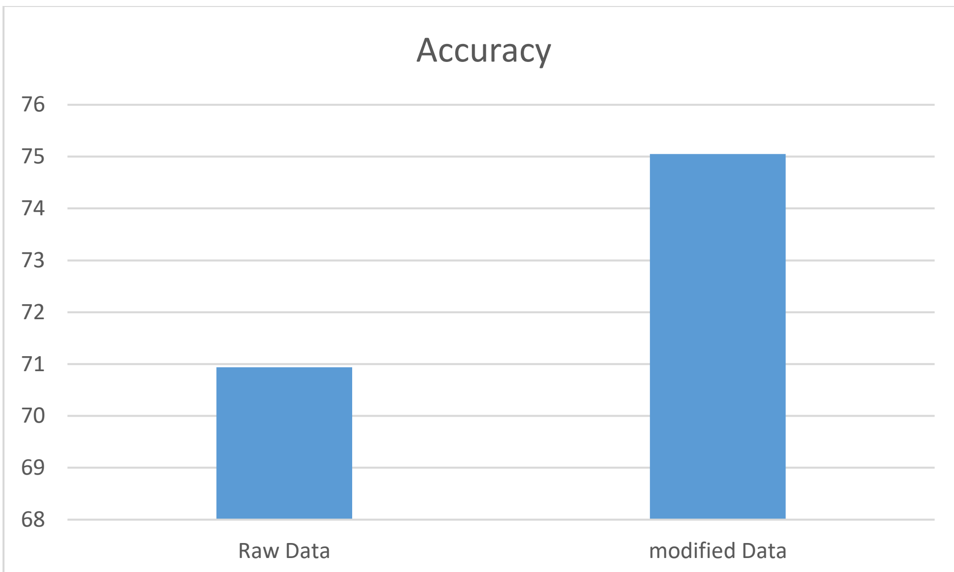


Figure 4.3 Accuracy graph for test dataset 1

Figure 4.3 show the difference of accuracy between the raw data model and the modified data model. From raw data model, we have achieved 70.94% accuracy and from modified data model we have achieved 75.05% accuracy. In case of test dataset1 we have increased 4.11% accuracy.

Test Dataset 2

Total tested data 948. Human face contains data 503 Non-human face 445

Confusion matrix for raw data model,

Table 4.5 Confusion matrix of raw data model for Test Dataset2

		Predicted	
		Human	Non-human
Actual	Human	**350**	**153**
	Non-human	**90**	**355**

Table 4.3 the confusion matrix shows the value of TP, TN, FP, and FN.

Here,

TP = 350, TN = 355, FP = 90, FN = 153

Table 4.6 Details Accuracy of raw data model for test dataset 2

Class	**value**
True positive rate	0.69
False positive rate	0.20
precession	0.79
recall	0.69
F1 measure	0.74

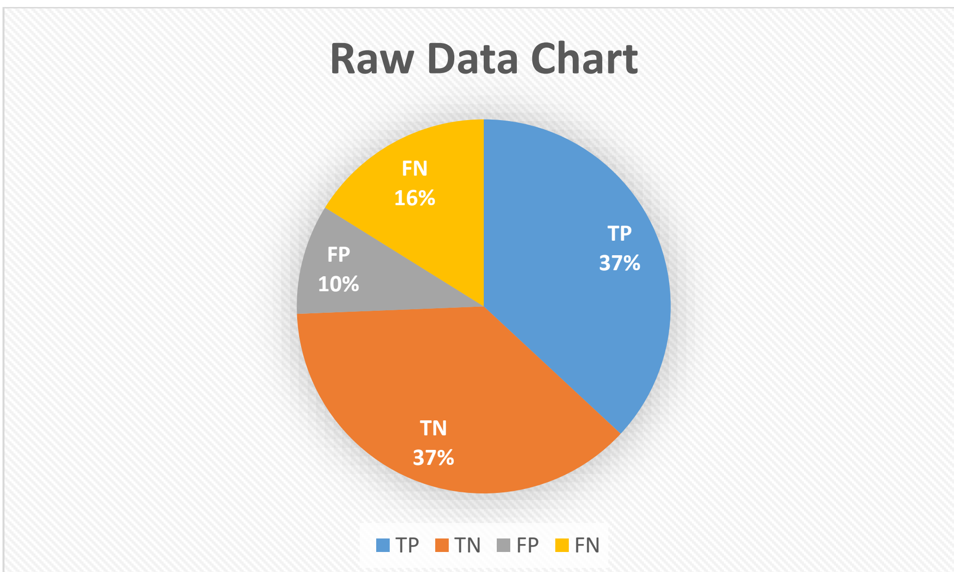


Figure 4.4 Pie chart of raw data model for training dataset 2

Figure 4.4 Shows that from raw data model we have achieved 37% TP and 37% TN in case of test dataset2. So, the raw data model can successfully detect 74% test data and failed to detect 26% test data.

From Raw data, we have achieved 74.37% accuracy.

Confusion matrix for modified data model,

Table 4.7 Confusion matrix of modified data model for Test Dataset2

		Predicted	
		Human	Non-human
Actual	Human	**380**	**123**
	Non-human	**104**	**341**

Table 4.7 the confusion matrix shows the value of TP, TN, FP, and FN.

Here,

TP = 380, TN = 341, FP = 104, FN = 123

Detail accuracy by class for modified data model

Table 4.8 Details Accuracy of modified data model for test dataset 2

Class	**value**
True positive rate	0.76
False positive rate	0.23
precision	0.79
Recall	0.76
F1 measure	0.77

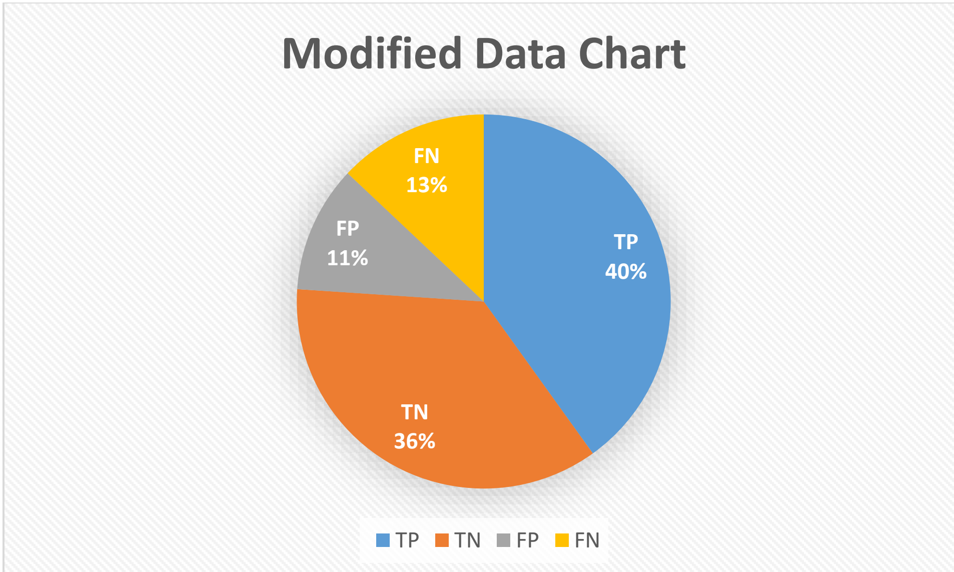


Figure 4.5 Pie chart of modified data model for training dataset 2

Figure 4.5 Shows that from modified data model we have achieved 40% TP and 36% TN in case of test dataset2. So, the modified data model can successfully detect 76% test data and failed to detect 24% test data.

From modified data, we have achieved 76.05% accuracy.

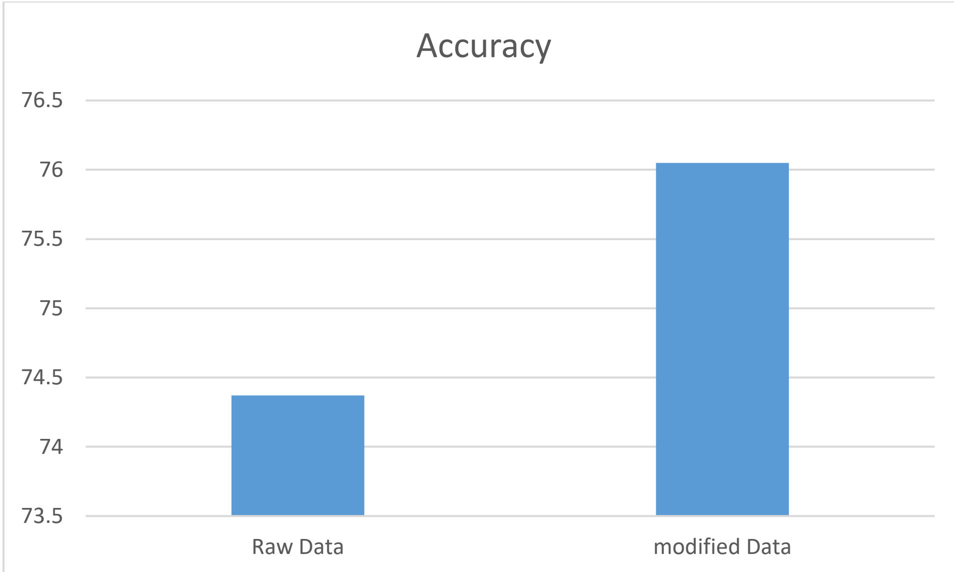


Figure 4.6 Accuracy graph for test dataset 2

Figure 4.6 show the difference of accuracy between the raw data model and the modified data model in case of test dataset2. From raw data model, we have achieved 74.37% accuracy and from modified data model we have achieved 76.05% accuracy. In case of test dataset2 we have increased 1.68% accuracy.

Test Dataset 3

Total tested data 648. Human face contains data 203 Non-human face 445

Confusion matrix for raw data model.

Table 4.9 Confusion matrix of raw data model for Test Dataset3

		Predicted	
		Human	Non-human
Actual	Human	**136**	**67**
	Non-human	**90**	**355**

Table 4.9 the confusion matrix shows the value of TP, TN, FP, and FN.

Here,

TP = 136, TN = 355, FP = 90, FN = 67

Detail accuracy by class for raw data model

Table 4.10 Details Accuracy of raw data model for test dataset3

Class	Value
True positive rate	0.67
False positive rate	0.20
precession	0.60
recall	0.67
F1 measure	0.63

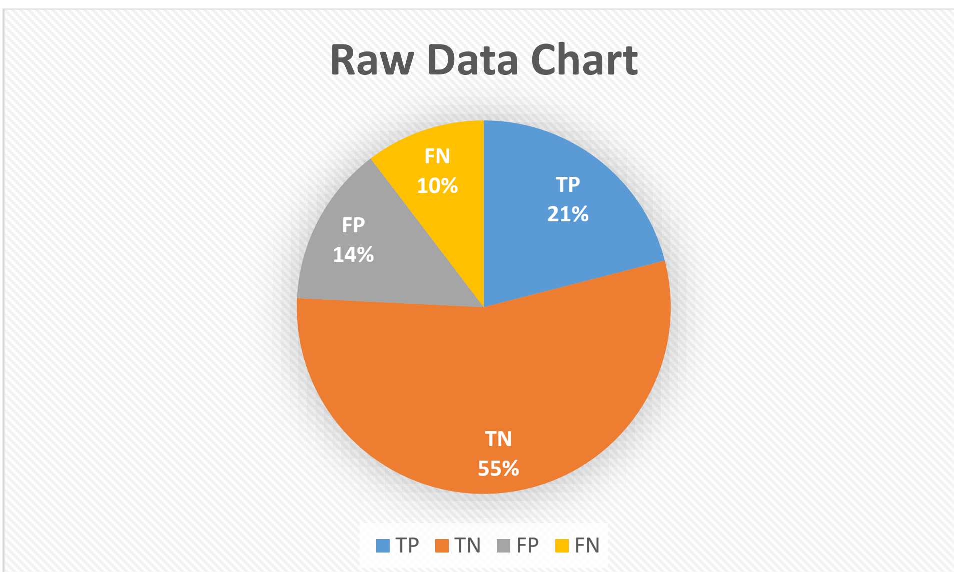


Figure 4.7 Pie chart of raw data model for training dataset 3

Figure 4.7 shows that from raw data model in case of test data3, we have achieved 21% TP and 55% TN. So, the raw data model can successfully detect 76% test data and failed to detect 24% test data.

From Raw data, we have achieved 75.77% accuracy

Confusion matrix for modified data model,

Table 4.11 Confusion matrix of modified data model for Test Dataset3

		Predicted	
		Human	Non-human
Actual	Human	151	**52**
	Non-human	**104**	**341**

Table 4.11 the confusion matrix shows the value of TP, TN, FP, and FN.

Here,

TP = 151, TN = 341, FP = 104, FN = 52

Detail accuracy by class for modified data model

Table 4.12 Details Accuracy of modified data model for test dataset3

Class	**value**
True positive rate	0.74
False positive rate	0.23
precision	0.59
Recall	0.74
F1 measure	0.66

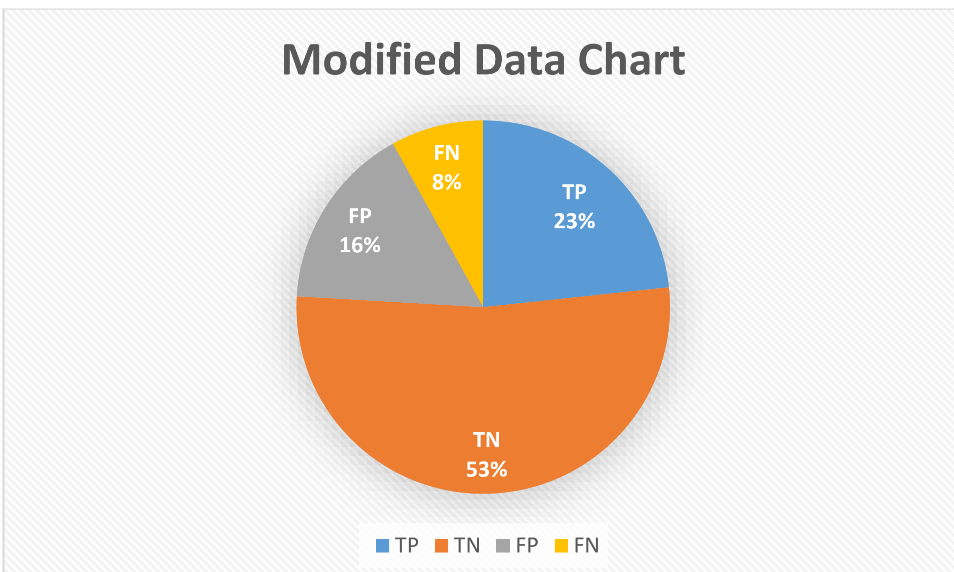


Figure 4.8 Pie chart of modified data model for training dataset 3

Figure 4.8 shows that from modified data model in case of test data3, we have achieved 23% TP and 53% TN. So, the raw data model can successfully detect 76% test data and failed to detect 24% test data.

From modified data, we have achieved 75.96% accuracy

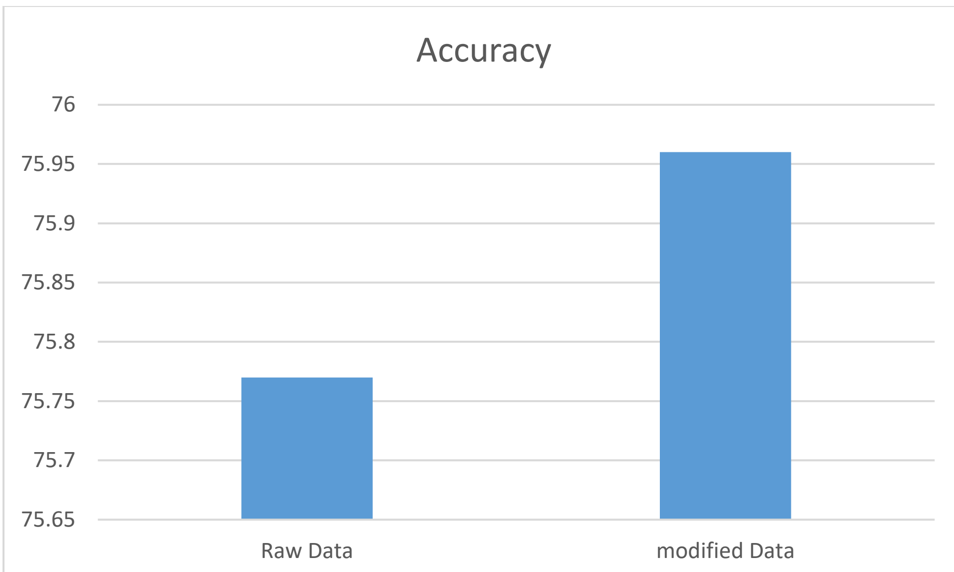


Figure 4.9 Accuracy graph for test dataset 3

Figure 4.9 show the difference of accuracy between the raw data model and the modified data model in case of test dataset3. From raw data model, we have achieved 75.77% accuracy and from modified data model we have achieved 75.96% accuracy. In case of test dataset3 we have increased .19% accuracy.

From test dataset3 we have increase 0.19% accuracy.

4.2 Performance Comparison

Comparison among three data set,

Table 4.13 Comparison amount three dataset.

Dataset	**Raw Data Accuracy**	**Modified Data Accuracy**
Test Dataset1	70.94%	75.05%
Test Dataset2	74.37%	76.05%
Test Dataset3	75.77%	75.96%

Table 4.13 shows the comparison of accuracy between test data model and modified data model. In all cases we have seen that the accuracy of modified data is greater than the accuracy of raw dataset. Is case of test dataset1 we have seen the comparison is greater than the other two test dataset. The reason is that in test dataset1 we use more positive image (human) than the negative image (non-human). We have modified only our positive raw images so that our network can extract more feature, as a result in test dataset when the number of positive image is large the accuracy deference will be more. In case of test dataset2 the positive image and negative is are almost same so the difference of accuracy between raw data model and modified data model is less than test dataset1. In case of Test dataset3 the positive image is less than the negative image so the accuracy difference is less than the test dataset2. In all three conditions, we have seen that the accuracy is increased when we use our modified data.

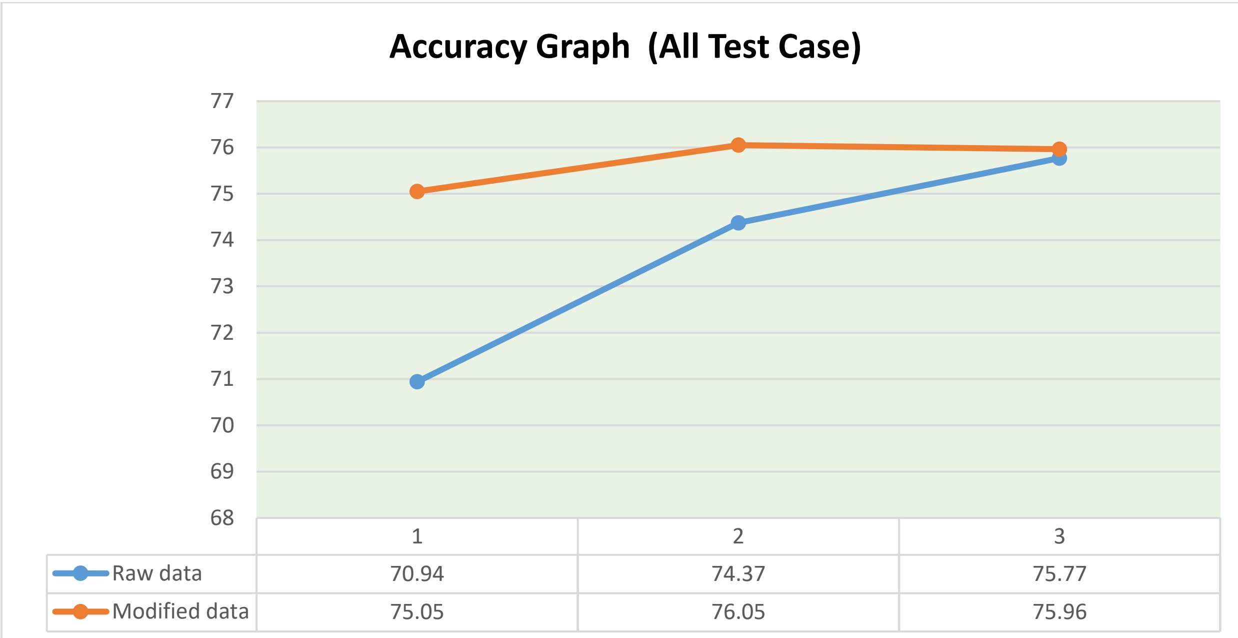


Figure 4.10 Comparison of three test dataset

Figure 4.10 shows the difference of accuracy between raw data and modified data getting from three types of test dataset. From test dataset1 we have achieved accuracy 70.94% for raw data model and 75.05% accuracy for modified data model we able to improve accuracy by 4.11%, from test dataset2 we have achieved accuracy 74.37% for raw data model and 76.05% for modified data model in case of test dataset2 we have achieved 1.68% improvement. From test dataset3 we have achieved 75.77% from raw data model and 75.96% accuracy for modified data model, in this case we improve accuracy by 19%.

4.3 Summary

In this chapter, we compare the results from three test datasets. We show how the accuracy increases for three type of test data. From test dataset1 we have achieved accuracy 70.94% for raw data model and 75.05% accuracy for modified data model we able to improve accuracy by 4.11%, from test dataset2 we have achieved accuracy 74.37% for raw data model and 76.05% for modified data model in case of test dataset2 we have achieved 1.68% improvement. From test dataset3 we have achieved 75.77% from raw data model and 75.96% accuracy for modified data model, in this case we improve accuracy by 19%. All types of test data we get improvement from raw data model to modified data model.

Chapter 5

Conclusion

5.1 Overall Conclusion

In this research, we have introduced a propose model of IoT based smart door using CNN classifier. The door will only open when any human stand in front of it. We have created a CNN model which can detect human and we also introduce a method that how we can increase the accuracy of any CNN model without increase the training dataset. To train a CNN network we need to collect a huge amount of data, in our case we need a large amount of positive (human) and negative (non-human) images. By using our method, we have shown how we can increase the accuracy without increasing the amount of training dataset. We have train our model with raw data and get 71% accuracy and then we retrain our model with our modified data and get 75% accuracy. We able to increase 4% of accuracy. This method can be applied in any CNN model, if any one builds a network which can detect dog or which can detect any object our method can be applied to get more accuracy.

5.2 Future Works

In future, we increase the amount out training dataset so that we could get more than 90% accuracy. We modify our CNN classifier that can recognize human faces so that we can use this model in many security surveillance and security related system where face detection needed.

References

[1] Shanshan Guo, Shiyu Chen and Yanjie Li (2016), Face Recognition Based on Convolutional Neural Network and Support Vector Machine. *International Conference on Information and Automation Ningbo, China, August 2016.* pp. 1788-1791

[2] Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt, Gang Hua, A convolutional neural network cascade for face detection *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, *MA, USA, June 2015.*

[3] Face Recognition Database by Massachusetts Institute of Technology [online] Available at : http://cbcl.mit.edu/software-datasets/heisele/facerecognition-database.html

[4] R. Vaillant, C. Monrocq, and Y. Le Cun. Original approach for the localisation of objects in images. IEE ProceedingsVision, Image and Signal Processing, 1994. 2

[5] Datasets by Visual Geometry Group, *Department of Engineering Science, University of Oxford* [online] Available at: http://www.robots.ox.ac.uk/~vgg/data3.html

[6] C. Garcia and M. Delakis. A neural architecture for fast and robust face detection. In Pattern Recognition, 2002. Proceedings. 16th International Conference on, 2002. 2

[7] Datasets for Computer Vision Research by Ponce Research Group [online] Available at : http://www-cvr.ai.uiuc.edu/ponce_grp/data//

[8] Caltech 10, 000 Web Faces by Caltech Vision Lab [online] Available at : http://www.vision.caltech.edu/Image_Datasets/Caltech_10K_WebFaces/

[9] LFWcrop Face Dataset [online] Available at: http://conradsanderson.id.au/lfwcrop/

[10] Fang Gao, Zhangqin Huang, Zheng Wang and Shulong Wang. An object detection acceleration framework based on low-power heterogeneous manycore architecture. *Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on, Dec. 2016*

[11] Il-kyu Hwang and Jin-wook Baek. Wireless access monitoring and control system based on digital door lock. *IEEE Transactions on Consumer Electronics (Volume: 53, Issue: 4, Nov. 2007). Nov. 2007*

[12] C. Szegedy, A. Toshev and D. Erhan, "Deep Neural Networks for object detection," Advances in Neural Information Processing Systems, pp. 2553-2561, 2013.

[13] Y. Li, S. Wang, Q. Tian and X. Ding, "Feature representation for statistical-learning-based object detection: *A review," PATTERN RECOGN, vol 48, no. 11, pp. 3542-3559, 2015.*

[14] Xi Li, Liming Zhao, Lina Wei, Ming-Hsuan Yang, Fei Wu, Yueting Zhuang, Haibin Ling and Jingdong Wang. DeepSaliency: Multi-Task Deep Neural Network Model for Salient Object Detection. *IEEE Transactions on Image Processing (Volume: 25, Issue: 8, Aug. 2016)*