B.TECH PROJECT REPORT

On

2D EAR BIOMETRICS USING DEEP LEARNING

by

SHIVAM PARASHAR
K GANESH RAJ

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

DECEMBER 2018

2D Ear Biometrics using Deep Learning

A PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degree
of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING
by

Shivam Parashar
K Ganesh Raj

Under the guidance of

Dr. Surya Prakash
Assistant Professor
Computer Science and Engineering

Department of Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY INDORE
December 2018

CANDIDATE'S DECLARATION

We hereby declare that the project entitled "2D Ear Biometrics on Deep Learning" submitted in partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science and Engineering, carried out under the supervision of Dr. Surya Prakash, Assistant Professor, Discipline of Computer Science and Engineering, IIT Indore is an authentic work. Further, we declare that we have not submitted this work for the award of any other degree elsewhere.

K Ganesh Raj

150001012

Shivam Parashar

150001033

CERTIFICATE BY BTP GUIDE

It is certified that the declaration made by the students is correct to the best of my knowledge and belief.

Dr. Surya Prakash,

Assistant Professor,

Discipline of Computer Science and Engineering,

IIT Indore

Preface

This report on “2D Ear Biometrics using Deep Learning” is prepared under the guidance of Dr. Surya Prakash.

Through this report, we have tried to use Deep Learning to create a Biometrics model that uses 2D Ear Images for identification and verification. We used a recurrent neural network for this purpose.

We documented the variation of data losses and Validation accuracy at each stage of training of the neural network. We have tried to the best of our abilities and knowledge to explain the content in a lucid manner. We have also added models and figures to make it more illustrative.

Place: IIT Indore

Date: 05/12/2018

Abstract

The problem statement we worked on was to implement a 2D Ear Biometrics system that uses Deep Learning for classification.

The use of Ear images for Biometric purposes is important in the field of biometrics because ear images are more unique than most other features and also are structurally stable at the same time. We implemented this using neural networks because neural networks use lesser space and once they are trained, they are also faster than most classification algorithms that do not use neural networks.

We have used SURF from OpenCV for the feature extraction part and keras, tensorflow libraries for the implementation of the neural network.

List of Contents

List of Figures

Chapter 1

Introduction

In an increasingly digital world, protecting confidential information is becoming more difficult. Traditional passwords and keys no longer provide enough security to ensure that data is kept out of the hands of hackers and unauthorized individuals. Additionally, with more devices and platforms connected to the Internet of Things, the need for ironclad security is paramount. This is where biometric security can transform the technology sector. Biometric authentication devices use unique traits or behavioral characteristics, such as fingerprints and voice recognition, to authenticate access to electronic assets. Because biometric information is unique to each person, fingerprint scans, for example, are an excellent way to ensure that the identification of users is sophisticated and complex enough.

Where passwords and physical tokens have fallen short, biometric authentication can succeed. Biometric authentication is an effective way to prove identity because it can't be replicated. Thanks to TouchID on smartphones, many consumers are already familiar with on-device biometrics.

Companies today are also realizing the benefits of biometric devices for protecting server rooms, work computers and other business assets. In a corporate environment, organizations need to make sure that unauthorized individuals are not allowed into secure systems. Additionally, for compliance reasons, companies need to ensure that workflow processes are followed correctly – certain employees

only have access to specific files. Using biometric scanners, companies can see each time a computer or server room is accessed and know who it was.

The human ear is a new feature in biometrics that has several merits over the more common face, fingerprint and iris biometrics. Unlike the fingerprint and iris, it can be easily captured from a distance without a fully cooperative subject. Also, unlike a face, the ear has a relatively stable structure that does not change much with the age and facial expressions.

	Face	Fingerprint	Hand geometry	Iris	Palm print
Universality	High	Medium	Medium	High	Medium
Uniqueness	Low	High	Medium	High	High
Permanence	Medium	High	Medium	High	High
Collectability	High	Medium	High	Medium	Medium
Performance	Low	High	Medium	High	High
Acceptability	High	Medium	Medium	Low	Medium
Circumvention	High	Medium	Medium	High	Medium

Figure 1.1: Perception of five common biometrics technologies. This table has been taken from 3D Biometric Systems and Applications, David Zhang · Guangming Lu.

1.1 Motivation for the work

The main motivation was to utilize the advantages of deep learning techniques create a model that identifies a person using a 2D image of his/her ear given that the model has already been trained with the training dataset consisting of ear images of whomsoever is to be identified.

There are already fingerprint scanners on almost every smartphone today owing to the major advancements in biometrics technology. But using fingerprints for biometrics is not always effective as some people do not have fingerprints and sometimes they get erased when subjected to physical abrasion. Another popular method of biometrics is Face recognition. But this fails when subjected to twins, lookalikes and in some cases even siblings. So Ear Biometrics proves to be a feasible alternative to currently more popular biometric methods as it is universal and is also easily collectable.

1.2 Why Ear Biometrics with Deep Learning?

We have chosen to implement the Ear Biometrics using Deep Learning techniques for our project because of the following key advantages:

- Unlike other biometrics like face recognition or fingerprint scans, ear images are easy to collect and do not have to be updated for a very long time.
- Every person's ear image has a number of unique edges and curves which when extracted using appropriate methods can create a unique feature set that can be utilized to identify them by Ear Biometrics systems using Deep Learning.
- Once trained with the dataset of the person to be identified the classification happens really fast in a neural network than most other biometric classifier algorithms.

Chapter 2

Ear recognition using HOG, Bag of Visual Words and Canny edge detection

We have worked on some other methods for the feature extraction part before we finally approached the current one. These methods weren't as effective as the final one.

Our previously implemented methods were:

- HOG – Histogram of Orientations
- Geometrical features with canny edge detection
- Bag of Features with SIFT feature extractor

2.1 HOG- Histogram of Orientations

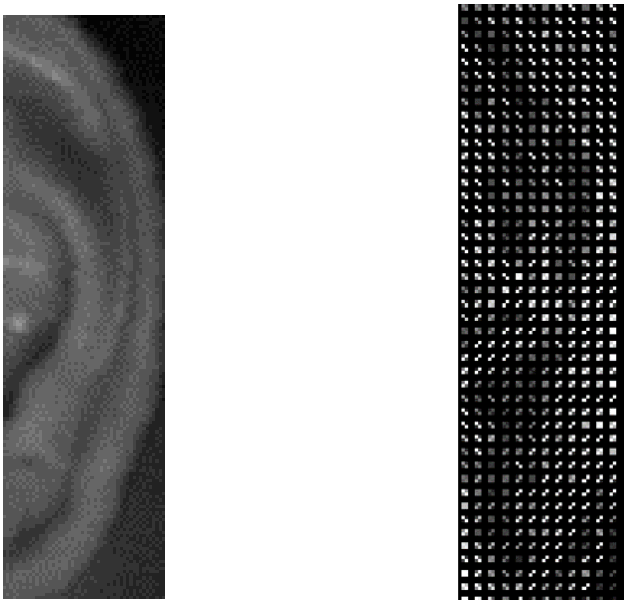
The images are initially pre-processed in four steps

- Binarization – The images are binarized to obtain contours of the ear.
- Mask Multiplication –The binary mask generated is then used to isolate the ear shape from the initial image.

- Dilation –The resultant image after application of mask is subjected to dilation.
- Noise Cancellation –The unnecessary patches in the image are removed in this step. This is done by removing all the clusters of pixels which are smaller than a threshold size.

The final pre-processed image is used to generate a histogram of oriented gradients (HOG). To calculate a HOG descriptor, we first calculate the horizontal and vertical gradients and then the Histogram of Gradients is generated [7]. We use windows of size 4x4 to generate a Histogram of 8 BINs. Each bin at an interval of 45 degrees (360/8) ranging from 0 to 360 degrees.

The obtained map of Histograms is subjected to 30 convolution layers of filter size 3 x 3 each. The generated feature maps are subjected to max pooling with a filter size of 2x2. The resultant reduced feature maps are then fed to a fully connected neural network that contains neurons with ReLU activation function. The accuracy obtained in this method based on the number of images accepted was 77.8%. We have used the IIT Delhi database for this method with 30 subjects.



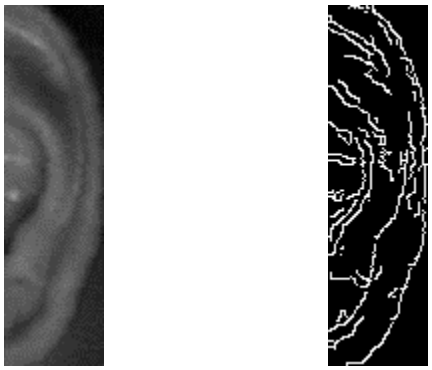
(a) (b)

Figure 2.1: an ear image (a) converted to a HOG feature map (b)

2.2 Geometrical Features with Canny edge detection

The geometrical features of an image are extracted by creating an edge map of that image using Canny edge detection. Before subjecting to edge detection, the image is preprocessed in four steps [2]:

- Histogram equalization: This is to increase the contrast of the image by adjusting the histogram distribution.
- Median Blur: It movies through the image by a sliding window and replaces the window by its median thereby reducing the noise in the image.
- Binarization: The image is then binarized with a suitable threshold.
- Opening and closing: The binarized image is subjected to morphological operations to remove noise and unnecessary blobs.



(a) (b)

Figure 2.2: (a) ear image. (b) Canny edge map

We finally apply canny edge detection on the final binarised images. It uses hysteresis thresholding. Any edges with intensity gradient more than *maxVa*l

are sure to be edges and those below *minVal* are sure to be non-edges, so discarded. Those who lie between these two thresholds are classified edges or non-edges based on their connectivity.

After the edge map is generated, the following geometrical features are extracted to create a feature set:

- X coordinates of centroids of 4 different quadrants in the edge image.
- Y coordinates of centroids of 4 different quadrants in the edge image.
- Calculate the distance between points and take the max distance from the sorted array.
- X coordinate of the centroid of the edge image
- Y coordinate of the centroid of the edge image

The feature sets are then input to a Convolutional Neural Network. The obtained accuracy with this method was 83%. We have used the IIT Delhi database for this method with 30 subjects.

2.3 Bag of Features with SIFT feature extractor:

Bag of Visual Words (BOVW) or Bag of Features (BoF) is the method of representing an image as a set of features. Features consist of key points and descriptors. Key points are the “stand out” points in an image, so no matter the image is rotated, shrink, or expand, its key points will always be the same. And descriptor is the description of the key point. We use the key points and descriptors to construct vocabularies and represent each image as a frequency histogram of features that are in the image. From the frequency histogram, later, we can find other similar images or predict the category of the image [4].

Since images do not actually contain discrete words, we first construct a "vocabulary" of SIFT [5] features representative of each image category. The features are then subjected to k-means clustering to create a visual vocabulary containing. The number of clusters, in this case, is 50.

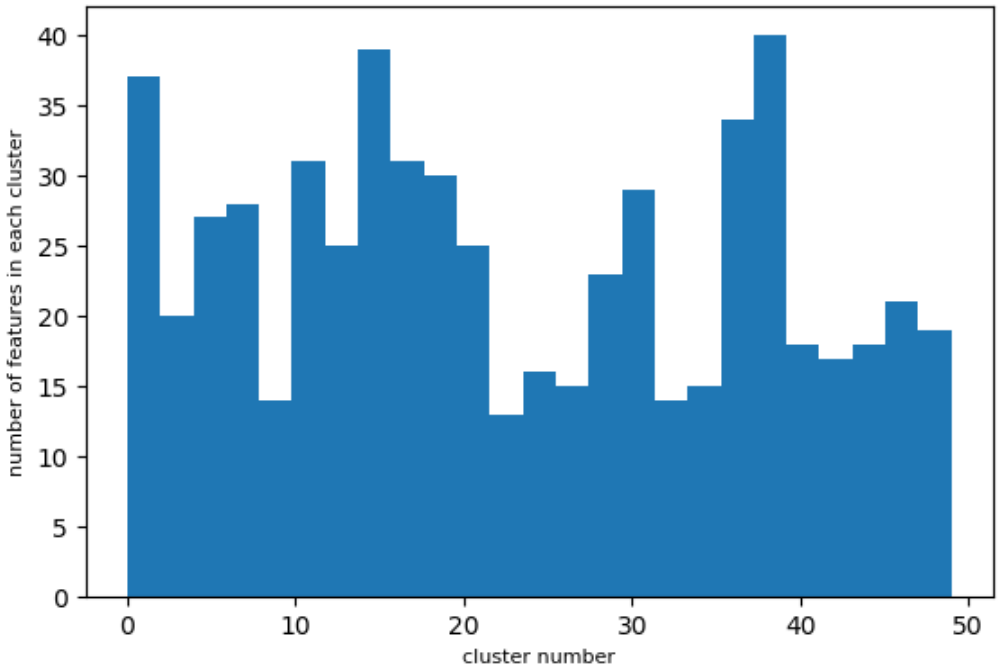


Figure 2.3: Histogram generated by k – means clustering of the features.

Chapter 3

Overview of the model

3.1 Parts of the model

Our model of 2D Ear Biometrics using Deep Learning is mainly segmented into the following four parts:

1. Data acquisition
2. Feature extraction
3. Creating and training the neural network
4. Testing the model

- Data Acquisition: We have used the 2D Ear images from the database we acquired. This database has 2D ear images of 50 subjects in total and 15 images of each subject. We have also used IIT Delhi database for previous methods we tried before concluding on the final model.
- Feature Extraction: We use SURF functionality from OpenCV on a python environment for this purpose. SURF stands for Speeded-up Robust Features. It is an algorithm which extracts some unique key points and descriptors from an image. A set of SURF key points and descriptors can be

extracted from an image and then used later to detect the same image. SURF uses an intermediate image representation called Integral Image, which is computed from the input image and is used to speed up the calculations in any rectangular area. It is formed by summing up the pixel values of the (x,y). More details about SURF will be explained in further chapters.

- Creating and training the neural network: For this project, we have used a class of neural network called the recurrent neural network. A recurrent neural network (RNN) is a class of artificial neural network where connections between nodes form a directed graph along a sequence. This allows it to exhibit temporal dynamic behavior for a time sequence. We have used a special type of RNN called the bidirectional LSTM. A bidirectional LSTM has two networks, one accesses information in a forward direction and another access in the reverse direction (as shown in the figure below). These networks have access to the past as well as the future information and hence the output is generated from both the past and future context. The neural network has been created using tensorflow libraries with keras [8] in the backend on a python environment.
- Testing and Validation: The trained model is then subjected to testing and validation. This is the part where the accuracy and the working status of the model are tested. The model we have used is a supervised neural network so the training and the testing data are labeled before they are given as input to the neural network and the results are documented accordingly.

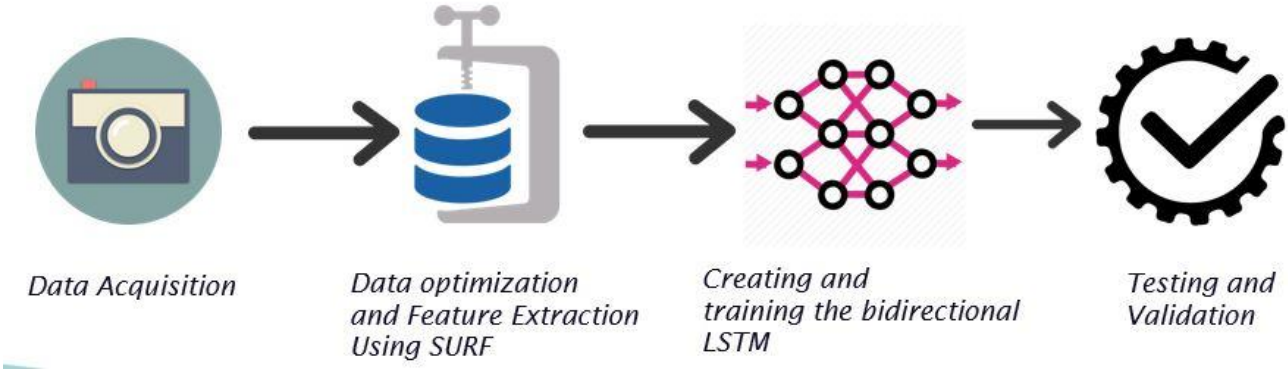


Figure 3.1: Overview of the model.

Chapter 4

Feature Extraction and SURF

4.1 Feature Extraction

Feature extraction involves reducing the number of resources required to describe a large set of data. When performing analysis of complex data one of the major problems stems from the number of variables involved. Analysis with a large number of variables generally requires a large amount of memory and computation power, also it may cause a classification algorithm to overfit to training samples and generalize poorly to new samples.

4.2 SURF

SURF stands for Speeded up Robust Features. It is an algorithm which extracts some unique key points and descriptors from an image. A set of SURF key points and descriptors can be extracted from an image and then used later to detect the same image. SURF uses an intermediate image representation called Integral Image, which is computed from the input image and is used to speed up the calculations in any rectangular area. It is formed by summing up the pixel values of the (x, y) coordinates from origin to the end of the image. This makes computation time invariant to change in size and is particularly useful while encountering large images. The SURF detector is based on the determinant of the Hessian matrix. The SURF descriptor describes how pixel intensities are distributed within a scale-dependent neighborhood of each interest point detected by Fast Hessian.

For orientation assignment, SURF uses wavelet responses in a horizontal and vertical direction for a neighborhood of size 6s. Adequate gaussian weights are also applied to it. Then they are plotted in a space as given in below image. The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window of angle 60 degrees. Interesting thing is that wavelet response can be found out using integral images very easily at any scale. [1]

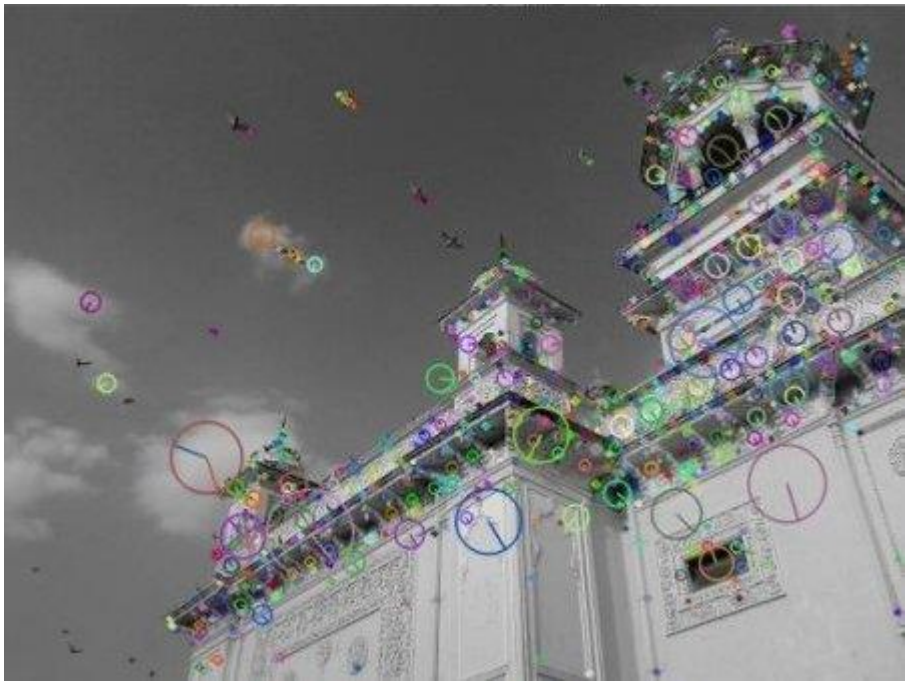


Figure 4.1: An image with SURF features and the direction labeled. Image has been taken from python- OpenCV official documentation.

For more distinctiveness, SURF feature descriptor has an extended 128 dimension version. The sums of (dx) and (|dy|) are computed separately for (dy < 0) and (dy >= 0). Similarly, the sums of (dy) and (|dy|) are split up according to the sign of (dx), thereby doubling the number of features. It doesn't add much computation complexity. OpenCV supports both by setting the value of flag **extended** with 0 and 1 for 64-dim and 128-dim respectively (default is 128-dim)

Another important advantage of SURF over other algorithms is the use of the sign of Laplacian (trace of Hessian Matrix) for underlying interest point. It adds no computation cost since it is already computed during detection. The sign of the Laplacian distinguishes bright blobs on dark backgrounds from the reverse situation. In the matching stage, we only compare features if they have the same type of contrast (as shown in the image below). This minimal information allows for faster matching, without reducing the descriptor's performance.

Chapter 5

Recurrent Neural Networks (RNN)

5.1 What is a Recurrent Neural Network?

A recurrent neural network (RNN) is a class of artificial neural network where connections between nodes form a directed graph along a sequence. This allows it to exhibit temporal dynamic behavior for a time sequence. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs [3].

RNN is the first Neural Network algorithm that remembers its input, due to an internal memory, which makes it perfectly suited for Machine Learning problems that involve sequential data. It is one of the algorithms behind the scenes of the amazing achievements of Deep Learning in the past few years.

A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor.

Recurrent Neural Networks accept an input vector x and give an output vector y. However, the differentiating feature of RNN is this output vector's contents are influenced not only by the input you just fed in but also on the entire history of inputs you've fed in in the past, unlike regular Feed Forward Neural

Networks. The RNN class has an internal state that it gets to update every time step is called. In the simplest case, this state consists of a single hidden vector that gives it the memory functionality.

5.2 Bidirectional LSTM model

LSTM stands for Long Short Term Memory. The disadvantage with RNN is that as the time steps increase, it fails to derive context from time steps which are much far behind. Enhancing the repeating module enables the LSTM network to remember long-term dependencies. Bidirectional LSTM has two networks, one access information in a forward direction and another access in the reverse direction (as shown in the figure below). The repeating module of a bidirectional LSTM provides three operations – Forget gate, Update gate and Output gate. These networks have access to the past as well as the future information and hence the output is generated from both the past and future context.

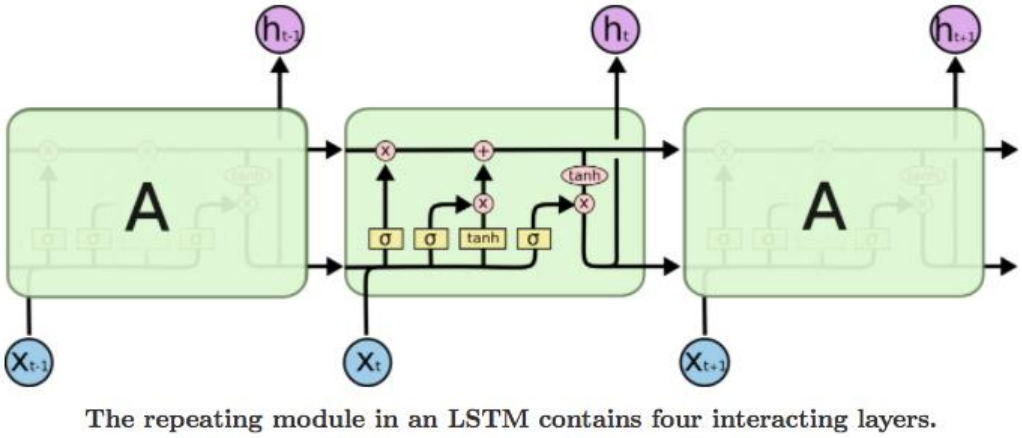


Figure 5.1: A repeating module in bidirectional LSTM. Image has been taken from An Introduction to Recurrent Neural Networks, a blog by Suvra Banerjee.

Chapter 6

Results

We have plotted the variation of testing and training accuracy as the model is being trained and as each epoch is being executed.

6.1 Loss function in neural networks

Each cycle in a neural network is called an epoch. An epoch is basically a forward pass and a backward pass of all the training data through the neural network.

Loss functions measure the mean squared or absolute error between a network's output and some target or desired output. Higher value of the loss function implies that the model is poorly trained and vice versa. The main objective in a learning model is to reduce (minimize) the loss function's value with respect to the model's parameters by changing the weight vector values through different optimization methods. Loss value implies how well or poorly a certain model behaves after each iteration/epoch of optimization.

6.2 Accuracy in neural networks

The accuracy of a model is determined at each epoch after the model parameters are updated as the learning takes place. Then the test samples are fed to the model and the number of mistakes (zero-one loss) the model makes are recorded, after comparison to the true targets. Then the percentage of misclassification is calculated. For instance, if the number of test samples is 1000 and model classifies 952 of those correctly, then the model's accuracy is 95.2%.

The accuracy here indicates the percentage of number of images matched out of all the images passed through the neural network. Initially the accuracy was low implying that the model's weights need to be updated to be able to identify the test images. As the model is being trained the weights are being updated according to the training images' features and the matching process becomes easier. Finally, as the neural network is being trained the accuracy increases till it reaches a maximum. Notice that the training accuracy is slightly higher than the testing accuracy. This indicates that training this model beyond a certain number of cycles will lead to a condition called overfitting where the model is trained more than it should be and it memorizes the training feature sets.

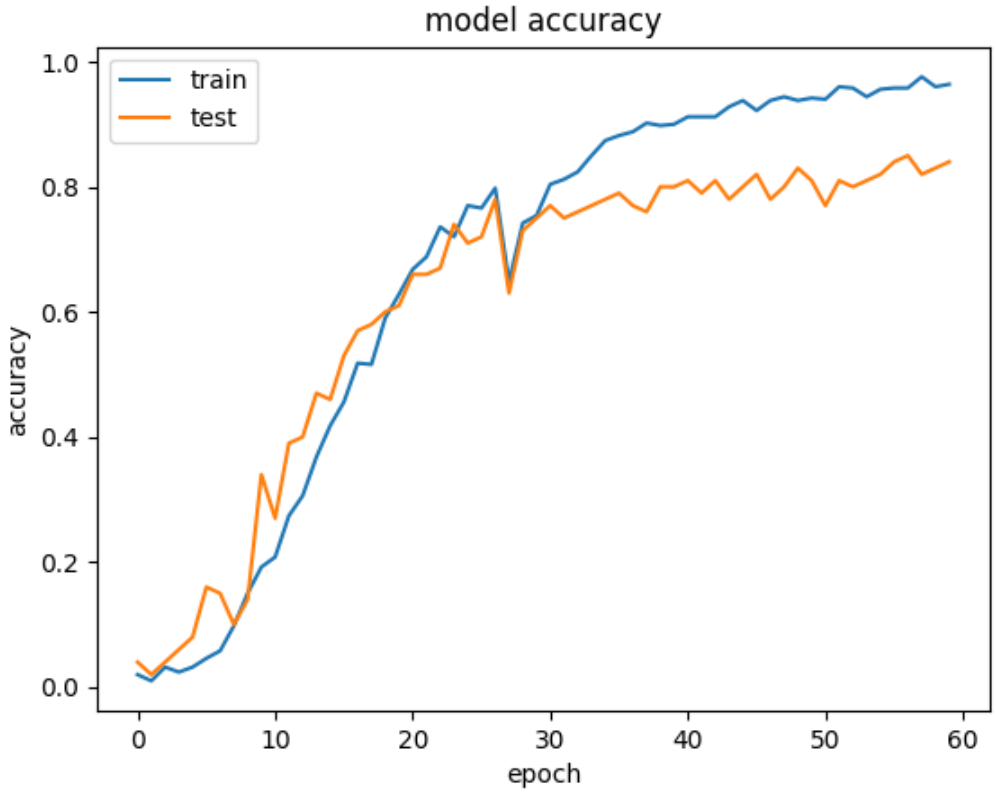


Figure 6.2: Variation of training and validation accuracy as the model is being trained.

The accuracy obtained with 50 subjects was 84%. We have captured photos of 50 subjects for this. The camera used was Nikon D5300, 24.2MP. We have used 10 training images for each subject and 2 testing images.

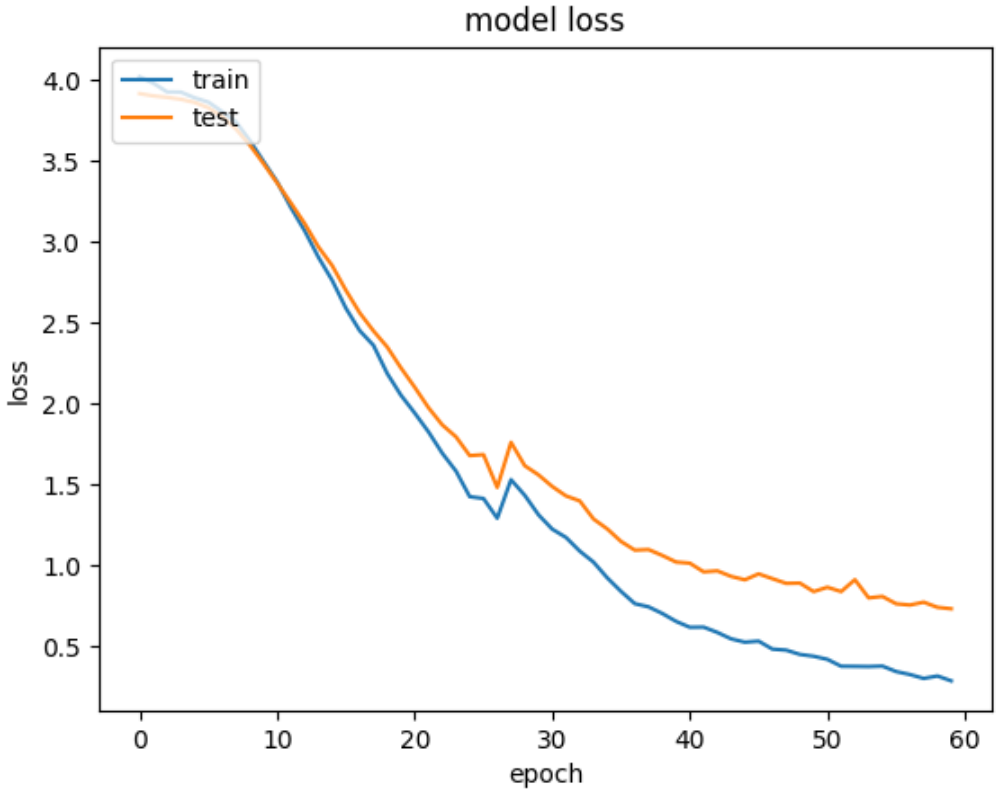


Figure 6.1: Variation of training and validation loss as the model is being trained.

Figure 6.1: Images of 3 subjects among 50 others, captured using a Nikon D5300 camera, 24,2MP.

6.3 Comparison with other methods

Method	**Accuracy obtained**
HOG feature extraction	77.8 %
Geometrical features using Canny edge detection	82.5 %
Bag of features(BOF) + KNN	79.8 %
Current Method- SURF with bidirectional LSTM	84.0%

Fig 6.2: Table of comparison with other methods

Chapter 7

Conclusion and future work

In this report, we have presented a method that uses neural networks to identify a person from a 2D image of his/her ear. With the increasing need for biometrics and due to the shortcomings of fingerprint and facial scanners, we think ear biometrics are going to become more and more popular in the future. Unlike fingerprint scanners or iris scanners, ear biometric system does not require any special sensors. It only requires a good camera which is common in most mobile phones today.

We have trained a deep learning model for identifying and authenticating users. The model has been trained and tested on TensorFlow with keras in the backend in an Ubuntu operating system. This work can be extended towards improving the accuracy and optimizing the time and space complexities of the model such that it becomes faster and easier to train and test.

References

[1] "Introduction to SURF (Speeded up Robust Features)". Accessed: 20-11-2018. [Online]. Available: https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_surf_intro/py_surf_intro.html

[2] "Human Ear Recognition Using Geometrical Features Extraction", Asmaa Sabet Anwara,d, Kareem Kamal A.Ghanyb,d, Hesham Elmahdyc, 2015 Faculty of Computers and Information, Cairo University, Cairo, Egypt.

[3] "Recurrent neural networks and bidirectional LSTM". accessed: 23-11-2018. [Online]. Available: https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5

[4] "Mathworks Documentation k-means clustering". accessed: 20-11-2018. [Online]. Available: https://in.mathworks.com/help/stats/kmeans.html

[5] "OpenCV python tutorials – Canny edge Detection". accessed: 17-10-2018. [Online]. Available: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_canny/py_canny.html

[6] "Introduction to SIFT (Scale-Invariant Feature Transform)". accessed: 15-11-2018. [Online]. Available: https://docs.opencv.org/3.1.0/da/df5/tutorial_py_sift_intro.html

[7] "Histogram of Oriented Gradients". Accessed: 12-12-2018. [Online]. Available: https://www.learnopencv.com/histogram-of-oriented-gradients/

[8] "The What's What of Keras and TensorFlow". accessed: 12-12-2018. [Online]. Available: https://www.upgrad.com/blog/the-whats-what-of-keras-and-tensorflow/