MAULANA AZAD
NATIONAL INSTITUTE OF TECHNOLOGY
BHOPAL, INDIA, 462003

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BIOMETRIC SYSTEM FOR IDENTIFICATION USING EEG SIGNALS

Major Project Report

Semester VIII

Submitted by:

SAUHARD PAREEK	191112237
SAMYAK MEHTA	191112213
CHEERLA SAI KRUSHEEL	191112259
KRISHNA KANT GOYAL	191112275

Under the Guidance of

Dr. MITUL KUMAR AHIRWAL

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Session: 2022-23

DECLARATION

We, hereby declare that the following report which is being presented in the Major Project Documentation Entitled as "**Biometric System For Identification Using EEG Signals**" is an authentic documentation of our own original work and to the best of our knowledge. The following project and its report, in part or whole, has not been presented or submitted by us for any purpose in any other institute or organization. Any contribution made to the research by others, with whom we have worked at Maulana Azad National Institute of Technology, Bhopal or elsewhere, is explicitly acknowledged in the report.

SAUHARD PAREEK	**191112237**
SAMYAK MEHTA	**191112213**
CHEERLA SAI KRUSHEEL	**191112259**
KRISHNA KANT GOYAL	**191112275**

MAULANA AZAD
NATIONAL INSTITUTE OF TECHNOLOGY
BHOPAL INDIA, 462003

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**Biometric System For Identification For Using EEG Signals**" submitted by:

SAUHARD PAREEK	**191112237**
SAMYAK MEHTA	**191112213**
CHEERLA SAI KRUSHEEL	**191112259**
KRISHNA KANT GOYAL	**191112275**

is the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering is the authentic work carried out by them under my supervision and guidance.

Dr. MITUL KUMAR AHIRWAL
(Minor Project Supervisor)

ACKNOWLEDGEMENT

With due respect, we express our deep sense of gratitude to our respected guide and coordinator **Dr. MITUL KUMAR AHIRWAL**, for his valuable help and guidance. We are thankful for the encouragement that he has given us in completing this project successfully.

It is imperative for us to mention the fact that the report of a major project could not have been accomplished without the periodic suggestions and advice of our project guide **Dr. MITUL KUMAR AHIRWAL** and project coordinators Prof. Bholanath Roy and Prof. Mansi Gyanchandani.

We are also grateful to our respected Hod, Dr. Deepak Singh Tomar for permitting us to utilize all the necessary facilities of the college.

ABSTRACT

This work aims to develop a biometric system for identification using Electroencephalogram as biometric modality. The biometric identification system are developed to prevent unauthorized access to privileged services. The manually extracted features has lot of variety and feature selection is required. So, a biometric system using 1-D CNN was developed for identification of users and convolutional layer was used for extracting features. The developed system was then used for authenticating the identity of the user. The dataset used for validating the performance of the developed system consists of 20 subject's data collected using 4 channels EEG. The accuracies for the different auditory stimuli were compared using the developed architecture.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Abbreviations

- PIN- Personal Identification Number
- EEG- Electroencephalogram
- SNR- Signal to Noise Ratio
- BCI- Brain Computer Interface
- LDC- Linear Discriminant Classifier
- GA- Genetic Algorithm
- CNN- Convolutional Neural Network
- LSTM- Long Short Term Memory
- STEW- Simultaneous Task EEG Workload
- MLP- Multi-Layer Perceptron
- KNN- K-Nearest Neighbor
- XG-Boost- Extreme Gradient Boosting
- RMS- Root Mean Square
- MSE- Mean Square Error
- EER- Equal Error Rate
- NN- Neural Network
- RC- Recall
- ACC- Accuracy

- PR- Precision
- F1- F1 score
- FAR- False Acceptance Rate
- FRR- False Rejection Rate
- HTER- Half Total Error Rate

Chapter1: Introduction

1.1. Biometric

Security is a necessity for human beings. So, with the evolution of technology, using it to strengthen the currently available security systems is essential. Biometric system, a one such security system, is constantly undergoing developments [1]. Biometric systems have marked their importance in variety of fields whether it be healthcare system, security like in police departments, banks and many more [2]. Biometric systems offer several advantages over traditional identification methods such as passwords, PINs, and ID cards. They provide a higher level of security, as it is difficult to fake or steal biometric data. They also provide convenience and speed, as biometric identification can be performed quickly and without the need for physical tokens or memorized information. Biometrics systems mainly deals with the problem of authentication and identification. Authentication, also known as verification, verifies whether the user is actually the one who it claims to be. So, the problem of authentication is a binary class problem, whether the person requesting is genuine or impostor. In Identification problem, the system tries to reveal the identity of the requesting person. The identification is basically a multiclass classification problem in which the system tries to know the identity of the requesting person [3]. The same data preprocessing, feature extraction and classifier design is shared by both identification and authentication [4].

The biometric system measures modalities associated with the identity of human beings to identify them [3]. Biometric modalities are physical or behavioral characteristics which are unique for each individual and help to establish their identity. Physical modalities usually do not require the subject to perform any specific activity, while the case is usually opposite for behavioral modalities. The physical modalities are affected by the interaction of the user during collection of samples. Hence, behavioral modality is usually superior to physical modality. EEG is also highly sensitive to mental tasks, therefore usually used as behavioral modality [5]. A variety of biometric modalities like fingerprint, iris, face have been used in biometric systems [6].

Each biometric modality has its own strengths and weaknesses, and different modalities may be more suitable for certain applications. For example, fingerprint recognition is commonly used for access control because it is highly accurate and relatively easy to use, while facial recognition is often used in law enforcement and surveillance because it can be used to identify individuals from a distance.

As technology continues to improve, new biometric modalities are also being developed, such as gait recognition, which uses an individual's unique walking pattern for identification. Ultimately, the choice of biometric modality will depend on the specific application and the level of security required.

However, biometric systems also raise concerns about privacy, security, and potential misuse of biometric data. It is therefore important to implement biometric systems carefully and with appropriate safeguards to protect the privacy and security of individuals.

Any biometric modality should have certain merits, which are mentioned below:

1) Universality – Every individual must possess these characteristics [5].

2) Distinctiveness - It must be different for 2 different individuals [7].

3) Permeance – The characteristics should not a vary a lot over a period [8].

4) Collectability – Data acquisition process must be simple [5].

5) Performance – The performance for the required application must be acceptable and the other factors required should be affordable [5].

6) Acceptability – It should be accepted by the participants [5].

7) Circumvention– The characteristic must be robust to attacks. They must be resistant to fraudulent techniques, so that it becomes difficult to spoof [8].

1.2. Problem Definition

Fingerprint, irises and voice etc. are popularly used as biometric modality but, they are vulnerable to attacks as the features can be extracted from high quality audio, video or image etc [6]. EEG, which expands to Electroencephalography, can also serve as biometric modality as it depends on person’s mood, brain structure, memory, and mental state, which make them a difficult target for attack [6] and acquisition is relatively less expensive in comparison to other brain imaging techniques [8]. Although, it has not been studied much due to the complicated nature of signal recording [9].

EEG must be looked as biometric modality, in terms of the merits, that it should possess which are universality, distinctiveness, permeance, collectability, performance, acceptability, and circumvention. EEG can be recorded from each and every person which is functional and living. So, it satisfies the property of universality. In the previously performed studies, it is seen that EEG has its roots linked to heredity and along with that EEG has been shown to possess high separability characteristics among individuals for groups of people which makes it a means that is distinct among different individuals. If it is looked in terms of permeance, if data recording expanding to several days, or several sessions, is used to train the classifier, there is reduction in recognition accuracy can be mitigated. Although, data collection using EEG is a quite cumbersome task, owing to the fact that it requires placing of electrodes around scalp, as per standards, for measuring electrical activity between neurons and, along with this, a conducting gel is usually used to reduce the resistance between the scalp and electrode to increase the SNR. But, now EEG systems which use elastic caps having electrodes attached, which allows the dynamic adjusting the location of electrodes. A new electrode type known as active electrode is now being manufactured. There is a built-in amplifier in this new type of electrode, which enhances the SNR and which removes the requirement to use the conducting gel. In terms of performance, the EEG has no long-term side effects and provides high accuracy when used for small scale problems. EEG recording has garnering acceptability among users after the spread of use of BCI technology. EEG cannot be easily be spoofed or forged in comparison to other types of biometric modalities such as fingerprint, or iris [5].

EEG requires very less amount of memory in comparison to image based biometric modalities like fingerprint, palm, face etc for storing the features. EEG, if used in combination with other biometric modality, will result in enhanced performance [5]. Thus, EEG may provide vast amount of application if used as biometric modality.

EEG possess the property of high time resolution which makes it possible to see brain dynamics. For an individual, the EEG is both specific and stable. EEG is most suitable for biometrics as compared to other biometric modalities as for it, the intra-personal difference is quite small and inter-personal difference is quite large. EEG can also detect that if the person is stressed or not, which can help us to ensure higher level of security, if required [2]. So, our problem is to develop a biometric identification system using EEG signal.

1.3. Research Gaps

The user identification using Electro encephalogram as biometric modality, is idea which is vastly unexplored. Most of the studies which have been carried out on user identification have limited themselves to the use of machine learning for identification. This approach requires features to be extracted manually which usually do not generalize well to all the new data. So, the use of CNN for feature extraction could help us to solve this issue of generalization. The studies that have used deep learning architecture for the identification, has limited themselves to resting state data usually. Our aim is to perform user identification using CNN model and explore effect of different type of auditory stimuli in user identification.

1.4. Objectives

The objectives of our work are:

- To explore EEG as biometric modality.
- To use CNN for feature extraction to solve the problem of generalization.
- To use Deep Learning architecture for identification.
- To explore the effect of different type of auditory stimuli in user identification.

1.5. Organization of Thesis

The remaining thesis is organised as follows: Chapter 2 represents Review of Literature. Chapter 3 represents Proposed Work Description. It explain Database, Pre Processing Data, System Overview and CNN Model. Chapter 4 represents Result Analysis and Description. It contains Training and testing and, Results and Parameters. Chapter 5 Summary, Conclusions and Future Directions. The last segment of our work is References.

Chapter 2: Review of Literature

Limited studies have been carried out on developing a biometric system with electroencephalogram as biometric modality. As the EEG signal acquisition process requires placing electrodes around scalp, which is a quite complex process, so the authors in [9], suggested the use of Genetic Algorithm and used the performance of LDC classifier as objective function. They have used LDC classifier for the evaluation of fitness function as it is fast and can compensate for the GA. The population was represented as binary strings, random in nature. Each chromosomes requires 61 bits to represent them. Each bit represents a channel. If the bit is 1, it means the channel is active and if the bit is 0, it means channel is inactive. Total 100 chromosomes were generated. Their fitness function covers the aspect of both classification performance and minimization of the channel but the weight given to minimization of channels is 0.5 in order to ensure that more importance is given to the performance. They used the data of 40 subjects originally collected using 61 channels and found that the identification performance does not decrease when 23 selected channels were used instead of all the 61 channels.

In [6], 1-D CNN LSTM neural network has been proposed for the user identification using EEG. In the enrolment phase, the biometrics for all the users are learnt and stored in the proposed network. The have performed batch normalization on the recorded signal and created the segments of 1-second recording. This normalized recording is then fed to the proposed CNN-LSTM architecture. The identity of the 1-second EEG signal is returned as final output. Their CNN-LSTM architecture consists of total 10 out of which first 4 are convolutional layers with number of filters 128, 256, 512, 1024 in order. The fifth layer is fully connected and dropout layer, with 192 neurons which is used to make the output of layer 4 to match layer 5 by reducing the number of features. Dropout layer helps in reducing overfitting and is therefore used. Both Layer 6 and layer 7 are the 192 neurons LSTM layers. For classification, layer 8 and layer 9 are used which are fully connected layers and for subject identification, SoftMax layer is used. The dataset sued consists of data of 109 subjects collected using 64 channels. The compared the performance of proposed architecture on the data of selected channel whose count were 4, 16, 32 and 64. The equal error rate and Rank-1 accuracy for 16 channel CNN-LSTM model has turned out to be 0.41% and 99.58% and there is no significant increase in performance is seen on increasing the channels. However, their accuracy fell to 94.28% when they used the proposed architecture on the data of selected 4 channels.

In [10], both user identification and authentication has been performed on resting state data recorded with eyes open and eyes closed, and they have leveraged the ability of CNN to extract features and have used Manhattan distance as matching algorithm. In enrolment phase, the acquisition of EEG signal is performed, and features vectors are extracted. These feature vectors are stored along with labels. Signals are acquired in base window of size 3 seconds. In identification phase, the signals are captured in 5 seconds of time window and then is split into overlapping segments using base window size. The extracted features from the segments are fused into one by averaging them. The fused feature vectors are matched to the stored features using matching algorithm which in their case is Manhattan distance. For identification, the identity given to the signal is the one to which the signal is most similar. For authentication, the person is considered genuine if the Manhattan distance for them is less than threshold or else it is considered impostor. They have used pyramidal architecture for feature extraction, in which number of filters for initial convolutional layers are more. They have evaluated the performance of 5 second 3 segments, 5 second 4 segments and 5 second 5 segments and found the best results on using 5 second 3 segment for both authentication and identification. They individually tested each channel for selecting the most effective channels for feature extraction and selected 2 channels. The dataset used by them also consists of EEG recordings from 109 subjects using 64 channels. However, they have not evaluated the system performance during different mental tasks.

In [3], they have used 3 datasets: Simultaneous Task EEG workload (STEW), EEG Alpha dataset and local dataset. In STEW, the data was recorded for 2.5 minutes with sampling frequency 128 Hz using 14 channels from 48 subjects in two sessions. In one session, they subject was in resting state and in other session, subject faced a mental task. In EEG ALPHA dataset, the data was collected from 20 subjects using 14 electrodes with sampling frequency 512Hz. Local dataset is been used in our work. The data from each dataset was segmented into non-overlapping segments of 4 second length. The dataset was filtered using 1-40-Hz first order Butterworth filter. They used 3 methods, namely, cluster map, Anova F-value, logistic regression weights for feature selection. They then fused the features resulted from above methods. For classification, they have used 3 different classifiers: MLP, KNN and XG-Boost. They have concluded that temporal threshold of 4 sec balances between the implement-ability and performance of a biometric authentication system. They found that feature selection shows a major reduction in computation time. They found that both, in ear phone and conducting headphone, differ in terms of performance and implement ability and found that performance is independent of language of the auditory stimuli. In [2], they have proposed the use of CNN for extracting features and conducting classification. The dataset was collected from 10 subjects with the subjects being in the resting state in 2 conditions i.e., eyes opened & eyes closed using 46 channels the sampling frequency 160Hz. They segmented data into 1 second & used 500 training samples and 50 testing samples. their architecture consists of 5 layers – Convolutional and Pooling layers being 2 each and 1 fully connected dense layer. Along with that they divided each 1 second resting state eyes opened EEG signal into several portions and also suggested that temporal threshold that can identify the subjects can be less than 200ms. They achieved the identification accuracy of 88% using the proposed model on resting state data with eyes open and closed for 10 subjects.

In [5], The dataset used was collected from 122 subjects, out of which 77 were control and 45 were alcoholic participants. There were either single or multiple two stimulus to which subjects were exposed. For each channel they used 1 feature which is the RMS value. The EEG signals were segmented into 1-second-long segments. The size of input layer depends on the number of RMS value i.e., 64. They performed classification using neural networks. In the first experiment, the data was considered from only control subjects were considered. The size of input layer is 64 units. The number of units in output layer is 45 as number of control subjects are 45. The hidden layer size was set to 200 neurons. The classifier was able to identify 42 out of 45 subjects correctly. The MSE (Mean square error) was 0.0043 for this experiment. In experiment 2, the data from all 122 subjects were considered and input layer size was 64 units and output layer size were 122 units. The hidden layer size after many trials was set to 500 neurons. They created a weighted connection between input and output layer which resulted in increasing the performance and they were able to identify 116 subjects correctly, the MSE value turned out to be 0.00186. For each channel, they suggested to consider the use of RMS spatial pattern as feature vector for user identification. They have built Neural network classifier which used radial basis and continuous tan, sigmoid activation function. The studies discussed above are briefly explained in table 1 below.

Table 1: BIOMETRIC STUDIES

S. No.	Subjects	Channels	Tasks/Relaxed	Recording duration	Model	Performance
1	109	64	Both	1 second	CNN-LSTM	Accuracy-99.58% EER-0.41% (16 channels]
2	109	64	Resting state	5 second	CNN, Manhattan distance	Accuracy-99.05% EER-0.187% (2 channels)
3	10	64	Resting state	200ms	CNN	Accuracy-88%
4	122	64	Task	1 second	RMS (feature), NN	113 correct out of 122
5	45	64	Task	1 second	NN	42 correct out of 45

Chapter 3: Proposed Work Description

The summarized steps of the work we performed as mentioned in the given Figure 1 below.

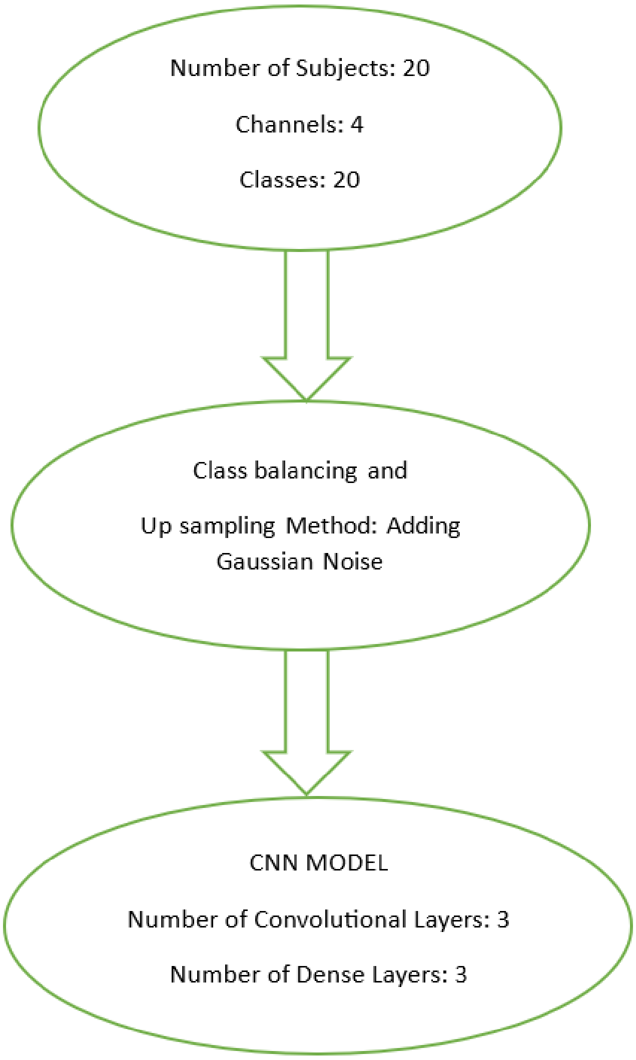

Figure 1: Flow Chart

3.1. Database

Data was collected through 4 electrodes which were placed at P4, F8, Cz and T7 positions following 10/10 International system. The sampling frequency was 200Hz. The reference and ground electrodes were placed at left and right ears. The subjects sat in a relaxed state on a chair. Recording for individual subjects were performed in a single day. The EEG signal acquisition was done, one, in relaxed state for 3 sessions with eyes open and closed and other, with auditory stimuli. For auditory stimulation, two types of ear phone were used, one, bone-conducting headphone and other, in-ear headphone. The signal acquisition for each headphone was performed in 3 conditions, depending upon the type of audio subject were made to listen, which were neutral music, native song and non-native song. For the Italian person, the non-native song was Arabic and for the Arabic person, it was Italian. The data provided consists of 2 minutes of recording involving the least amount of noise [3]. Since there are 20 subjects, hence number of classes is also 20.

3.2. Pre-Processing Data

The performance increase cannot justify the memory and computational resources required in case of long threshold, while performance is not optimal for short recordings. Hence, as suggested in [3], we picked temporal threshold as 4 seconds. As the sampling frequency is 200Hz. So, the samples that can be recorded in recording duration of 4 seconds is 800. There were mostly 30 instances for each subject. For each experiment, one file consists of data recorded through one channel for a subject. The dimensions of each such file were (instance, 801) where last column represents class. User 1 has been assigned label 0, user 2 has been assigned label 1, and so on and so forth. Number of samples for each subject corresponding to each experiment are mentioned in table 2.

Table 2: SUBJECT-WISE INSTANCE COUNT FOR ALL EXPERIMENTS

	In ear-native (Exp-5)	**In ear-non native (Exp-6)**	**In ear neutral (Exp-7)**	**Bone conducting native (Exp-8)**	**Bone conducting non native (Exp-9)**	**Bone conducting neutral (Exp-10)**
S01	30	30	30	30	30	30
S02	30	30	30	30	30	30
S03	30	29	55	30	30	30
S04	30	30	30	30	30	30
S05	50	30	30	30	30	30
S06	30	30	30	30	30	30

S07	30	30	30	30	30	30
S08	30	30	30	30	30	30
S09	30	30	30	30	30	30
S10	30	30	30	30	30	30
S11	30	30	30	30	30	30
S12	30	30	30	30	30	30
S13	30	30	30	30	30	30
S14	30	30	30	30	30	30
S15	30	30	30	30	30	30
S16	30	30	30	30	30	30
S17	30	30	30	30	30	30
S18	30	30	30	30	30	35
S19	30	30	30	30	30	30
S20	30	30	30	30	30	30

Then, the files prepared were uploaded on the google drive as Google Colab was used to support implementation. The files were read and were checked for null values or missing values. If found, we may need to delete the entire row or column, or we can replace it with the mean, median, or mode. In our case, no missing value was found. There was a issue of class imbalance. To handle this issue, we used the approach of gaussian noise. First, the class with maximum instance count was identified and instance count for that class was noted. Now, the new instances for all the other classes were generated to match the noted maximum instance count. For generating instances, the gaussian noise, with standard deviation 1 and mean 0, was added to the original instance. The noise was added to the original instance after multiplying it with multiplicative factor with value as 0.2. We produced the one new instance for each samples till the issue of class imbalance was resolved. The number of instance is quite small. So, the upsampling operation was performed to increase the number of samples. To upsample the data, the gaussian noise was added, with the standard deviation and mean being 0 and 1, respectively. We produced 10 new samples for each original sample, of which the first 5 samples were produced using a multiplicative value of 0.2 and the rest of the 5 samples were produced keeping the multiplicative value equal to 0.3. Along with the newly added samples, the original samples were also kept. This data was split into train, test, and validation, each of which was in 3-d form where each row represented the number of samples, each column represented timestamps, and the 3rd dimension represented the channel. We created a generating noise function to increase the number of samples.

3.2.1. Code

```
def balance(data_n, c):
  dfn  = pd.DataFrame()

  for row in range(0,c):
    mu = 0
    sigma = 1

    noise=np.random.normal(mu,sigma,size=data_n.iloc[[row]].shape)
    df  = pd.DataFrame()
    df = data_n.iloc[[row]] + 0.2*noise
    dfn = pd.concat([dfn,df], axis = 0, ignore_index = True)
  return dfn
```

3.2.2. Explanation

The function takes data frame and a variable c, representing the number of samples to be generated to balance the class, as input. Original instances were traversed in order and for each instance a new instance is generated by adding the gaussian to the original instance. The noise is not added completely; instead, it is multiplied by a value called the multiplicative value, which here is less than 0.5, and then added to the original sample. The multiplicative value, here, used is 0.2.

3.2.3. Code

```
def generatenoise(data_n):
  dfn  = pd.DataFrame()

  for row in range(len(data_n)):
    mu = 0
    sigma = 1
    for j in range(0,5):
      noise = np.random.normal(mu,sigma,size = data_n.iloc[[row]].shap
e)
      df  = pd.DataFrame()
      df = data_n.iloc[[row]] + 0.2*noise
      dfn = pd.concat([dfn,df], axis = 0, ignore_index = True)

    for j in range(0,5):
      noise = np.random.normal(mu,sigma,size = data_n.iloc[[row]].shap
e)
      df  = pd.DataFrame()
      df = data_n.iloc[[row]] + 0.3*noise
      dfn = pd.concat([dfn,df], axis = 0, ignore_index = True)

  return dfn
```

3.2.4. Explanation

This function takes a data frame as input. Then, the two loops are executed, the outer loop to select the sample that is to be up sampled. The inner loops are used to generate new samples. This is done by generating the gaussian noise and adding it to the original sample. The noise is not added completely; instead, it is multiplied by a value called the multiplicative value, which here is less than 0.5, and then added to the original sample. Since, the total number of new samples generated by this process for a sample is 10. The first inner loop generates the first 5 samples using a multiplicative value of equal to 0.2, and the second inner loop generates the remaining 5 samples, keeping the multiplicative value at 0.3. The number of instances for each subject corresponding to a experiments are mentioned in Table 3.

Table 3: SUBJECT-WISE INSTANCE COUNT FOR ALL EXPERIMENTS AFTER UPSAMPLING

	In ear-native (Exp-5)	In ear-non native (Exp-6)	In ear neutral (Exp-7)	Bone conducting native (Exp-8)	Bone conducting non native (Exp-9)	Bone conducting neutral (Exp-10)
S01	550	330	605	330	330	385
S02	550	330	605	330	330	385
S03	550	330	605	330	330	385
S04	550	330	605	330	330	385
S05	550	330	605	330	330	385
S06	550	330	605	330	330	385
S07	550	330	605	330	330	385
S08	550	330	605	330	330	385
S09	550	330	605	330	330	385
S10	550	330	605	330	330	385
S11	550	330	605	330	330	385
S12	550	330	605	330	330	385
S13	550	330	605	330	330	385
S14	550	330	605	330	330	385
S15	550	330	605	330	330	385

S16	550	330	605	330	330	385
S17	550	330	605	330	330	385
S18	550	330	605	330	330	385
S19	550	330	605	330	330	385
S20	550	330	605	330	330	385

3.3 System Overview

The biometric system consists of two phases:

1. Enrollment phase: In enrollment phase, the EEG biometrics of all 20 user's are learned, trained and stored in Convolutional neural network.

2. Development/Identification phase: In identification phase, the identity of the person to whom the EEG recording of 4 sec belongs is returned as output.

3. Testing phase/Authentication phase: In authentication phase, the identity of a person attempting to access the service is returned.

3.4 CNN Model

A 1-D CNN, or one-dimensional convolutional neural network, is a type of CNN that is commonly used for processing one-dimensional sequential data such as time series, audio signals, and text data. 1-D CNNs have been successfully applied in a wide range of applications, including speech recognition, music classification, and natural language processing. A 1-D CNN consists of two types of blocks, namely, convolutional blocks and output blocks. The convolutional block has a 1-Dimensional convolutional layer, a batch normalization layer, and a 1-Dimensional max pooling layer with some kind of activation function. In a 1-D CNN, the convolutional layers apply a set of filters to the input data, which slide along the time axis to capture local patterns and features. A convolutional layer is used for feature extraction by creating a kernel. This kernel gets convolved with the input layer and produces a feature. The number of feature maps generated is equal to number of filters. The architecture of a 1-D CNN typically includes several convolutional layers with increasing numbers of filters, interspersed with activation functions. A batch normalization layer is used to standardize the input values. It helps in increasing the performance of the model. The 1-D max pooling layer picks the maximum value from the window of size equal to the pool size. The output layer consists of a flatten layer and a dense layer with some kind of activation function. The dense layer classifies the output of the convolutional block. Like all other layers, the dense layer also contains neurons. It calculates the weighted average and passes it through the Relu/Softmax layer or any other kind of activation. And the final output is obtained from the last dense layer. They are particularly useful for processing sequential data because they can capture both short-term and long-term temporal dependencies in the data. This is illustrated through Figure 2 below.

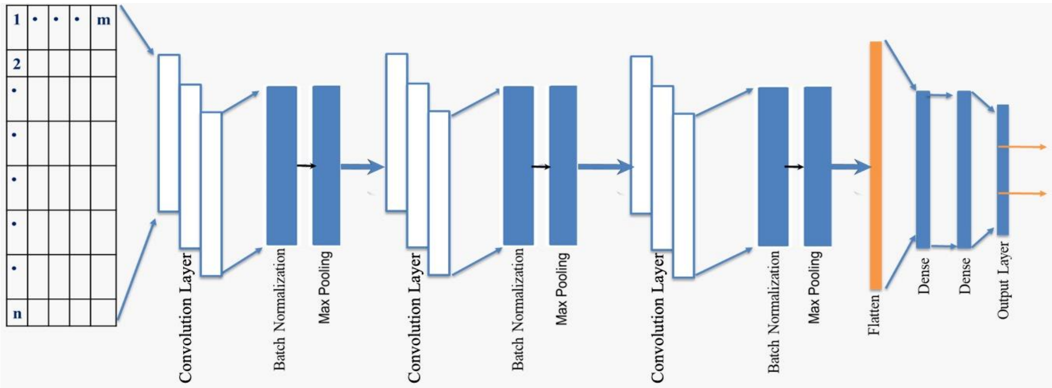

Figure 2: Schematic Representation of CNN

In the model, we have used 3 convolutional layers, 3 batch normalization layers, and 3 max pooling layers. In the output block, there are 3 dense layers. There is no padding provided in any of the convolutional layers, and the input shape has been changed accordingly. Also, the stride in the max pooling layer has been kept at 1.

3.4.1 Code

```
def build_model():
  model = tf.keras.models.Sequential()

  model.add(tf.keras.layers.Conv1D(filters = 64, kernel_size = 8, act
ivation = 'relu',  input_shape = (800,4)))
  model.add(tf.keras.layers.BatchNormalization())
  model.add(tf.keras.layers.MaxPooling1D(pool_size = (3), strides = (
1)))

  model.add(tf.keras.layers.Conv1D(filters = 128, kernel_size = 8, ac
tivation = 'relu', input_shape = (791,64)))
  model.add(tf.keras.layers.BatchNormalization())
  model.add(tf.keras.layers.MaxPooling1D(pool_size = (3), strides = (
1)))

  model.add(tf.keras.layers.Conv1D(filters = 256, kernel_size = 8, ac
tivation = 'relu',  input_shape = (782,128)))
  model.add(tf.keras.layers.BatchNormalization())
  model.add(tf.keras.layers.MaxPooling1D(pool_size = (3), strides = (
1)))

  model.add(tf.keras.layers.Flatten())

  model.add(tf.keras.layers.Dense(units = 64, activation= 'relu'))

  model.add(tf.keras.layers.Dense(units = 64, activation= 'relu'))

  model.add(tf.keras.layers.Dense(units = 20, activation= 'softmax'))

  model.compile(optimizer = 'Adam', loss = 'categorical_crossentropy'
, metrics = ['accuracy'])
  return model
model = build_model()
model.summary()
```

3.4.2. Explanation

Here, the build_model function is used to create the model, and there are 3 convolutional blocks, each with one convolutional layer, one batch normalization layer, and one max pooling layer. The convolutional layer does not possess any kind of padding and each layer has a kernel size of 8 and relu as its activation function, and for 3 convolutional layers, the filters are 64, 128, and 256, respectively, and the input shape is (800,4), (791,4), and (782,4), respectively. Each max pooling layer has a pool size of 3 with a stride of 1. Then, comes the output block, which contains three dense layers, of which dense layers 1 and 2 both use "relu" as an activation function, and both are of 64 units. The last dense layer, which is also the output layer, is made up of 20 units and uses the "softmax" layer as an activation function. Adam optimizer has been used with a categorical cross entropy loss function. This is listed in Table 4.

Table 4: MODEL ARCHITECTURE

Type of Layer	Output Shape	Other Parameters of each layer
Conv1D Batch Normalization Max Pooling 1D Activation	(793;64) (793;64) (791;64) (791;64)	Filters - 64, Kernel Size - 8, Pool Size – 3, Activation - Relu
Conv1D Batch Normalization Max Pooling 1D Activation	(784;128) (784;128) (782;128) (782;128)	Filters - 128, Kernel Size - 8, Pool Size – 3, Activation - Relu
Conv1D Batch Normalization Max Pooling 1D Activation	(775;256) (775;256) (773;256) (773;256)	Filters - 256, Kernel Size - 8, Pool Size – 3, Activation - Relu
Flatten Dense Dense Output	(197888) (64) (64) (20)	Units - 64, Activation – Relu Units - 64, Activation – Relu Units - 20, Activation – SoftMax

Chapter 4: Result Analysis and Description

4.1. Training and Testing

The up sampled data has been split into train and test using the train_test_split function with a split ratio of 0.25. The train data has been split further, into actual train data and validation data using the train_test_split function with a split ratio of 0.3. The training as well as learning of a model can be done using the Adam optimizer or any other kind of optimizer. The given optimizer uses a categorical cross entropy function and does the adaptation of weights. The model is trained with a fit function and evaluated using test data. The number of instances for each experiment after splitting the data into training, testing and validation is mentioned in table 5.

Table 5: TRAINING, TEST AND VALIDATION DATA

	Training Dataset	**Validation Dataset**	**Testing Dataset**
Exp-05	5768	2472	2760
Exp-06	3458	1482	1660
Exp-07	6342	2718	3040
Exp-08	3458	1482	1660
Exp-09	3458	1482	1660
Exp-10	4032	1728	1940

The comparison between training and validation accuracy and training and validation loss for experiment 5 is shown in Figure 3 and Figure 4, respectively.

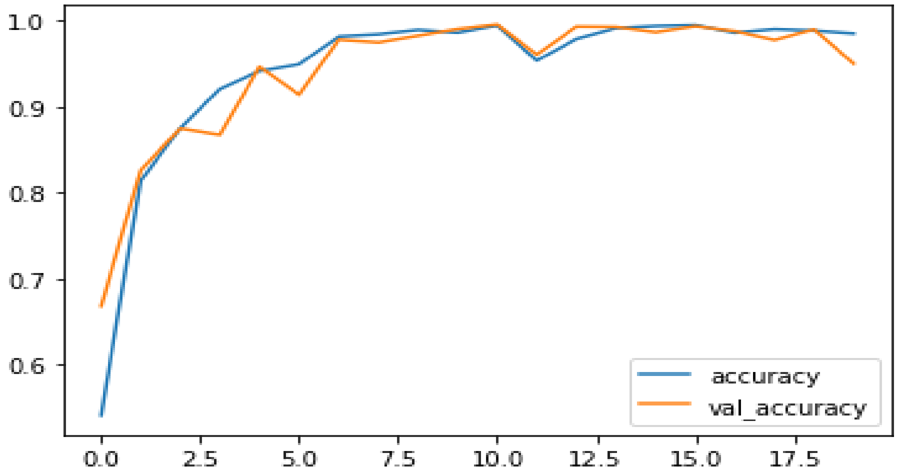

Figure 3: Comparing Accuracy And Val Accuracy for exp-5

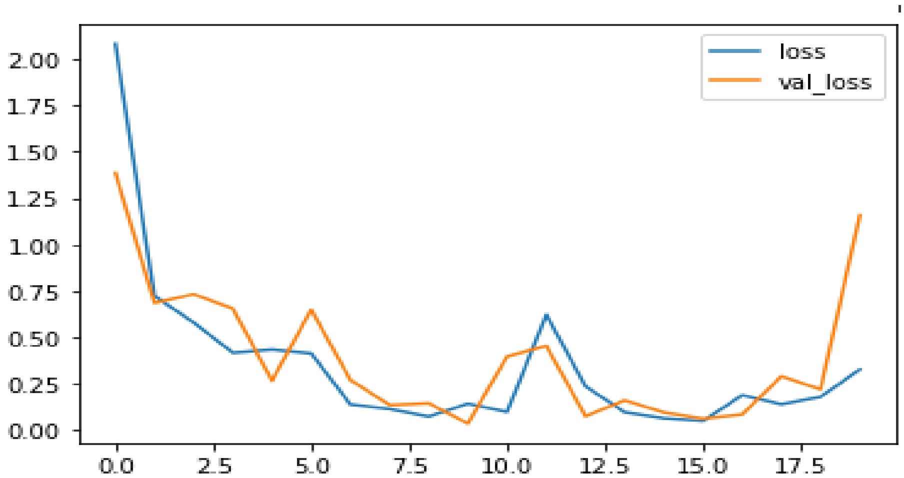

Figure 4: Comparing Loss And Val_Loss for exp-5

The comparison between training and validation accuracy and training and validation loss for experiment 6 is shown in Figure 5 and Figure 6, respectively.

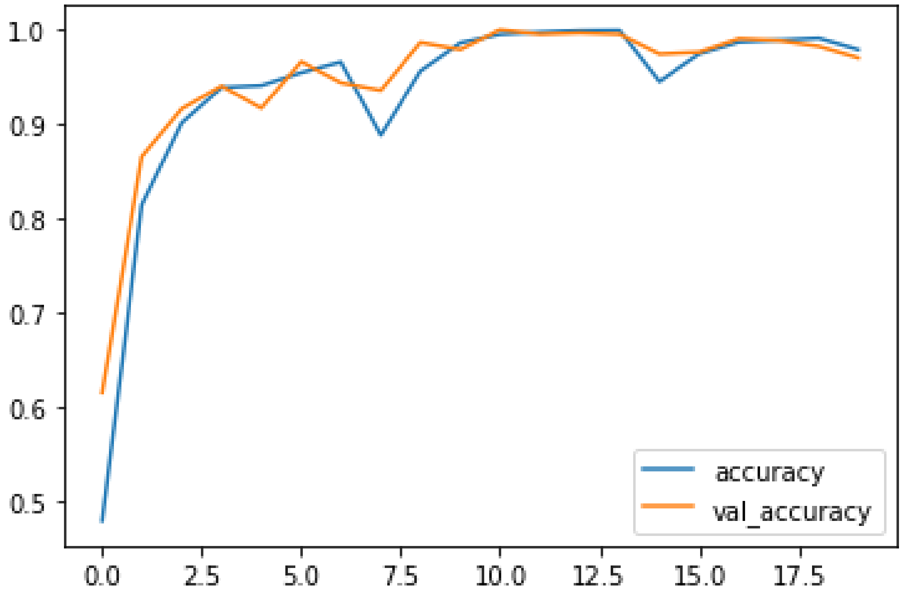

Figure 5: Comparing Accuarcy and Val Accuracy for exp-6

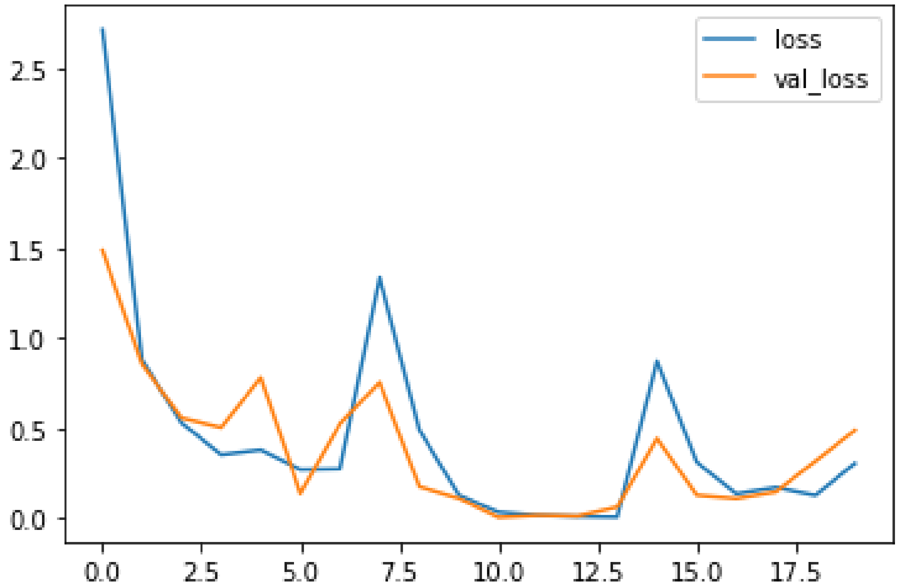

Figure 6: Comparing Loss and Val_Loss for exp-6

The comparison between training and validation accuracy and training and validation loss for experiment 7 is shown in Figure 7 and Figure 8, respectively.

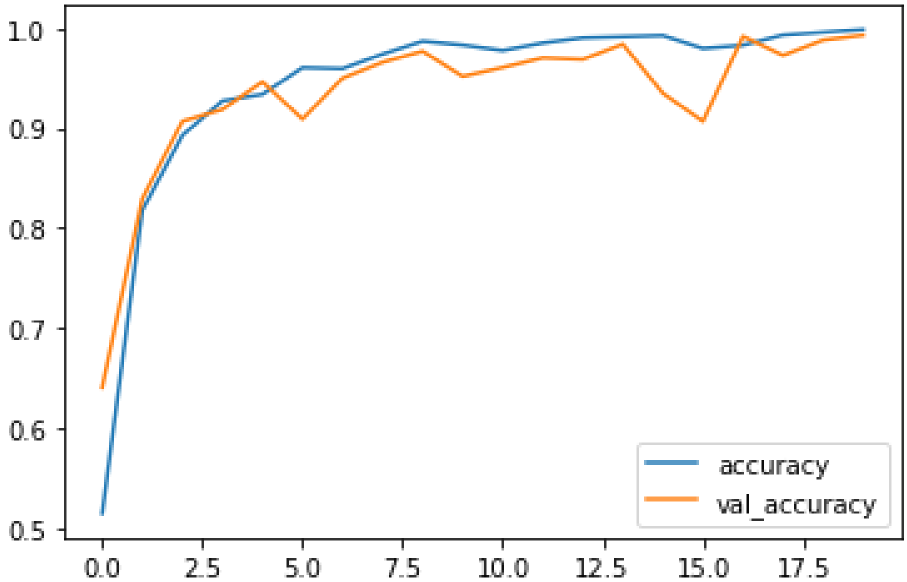

Figure 7: Comparing Accuarcy And Val Accuracy for exp-7

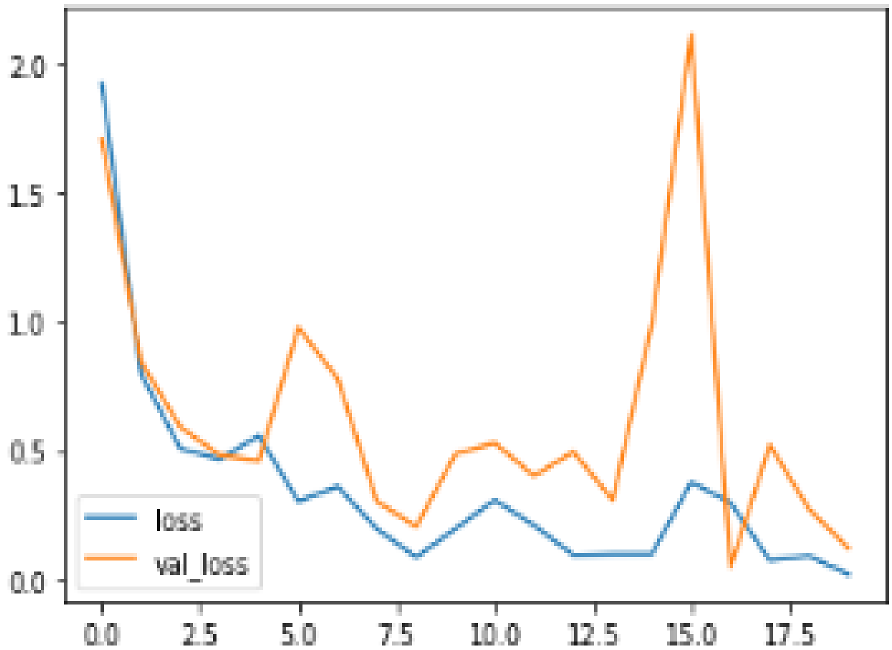

Figure 8: Comparig Loss and Val_Loss for exp-7

The comparison between training and validation accuracy and training and validation loss for experiment 8 is shown in Figure 9 and Figure 10, respectively.

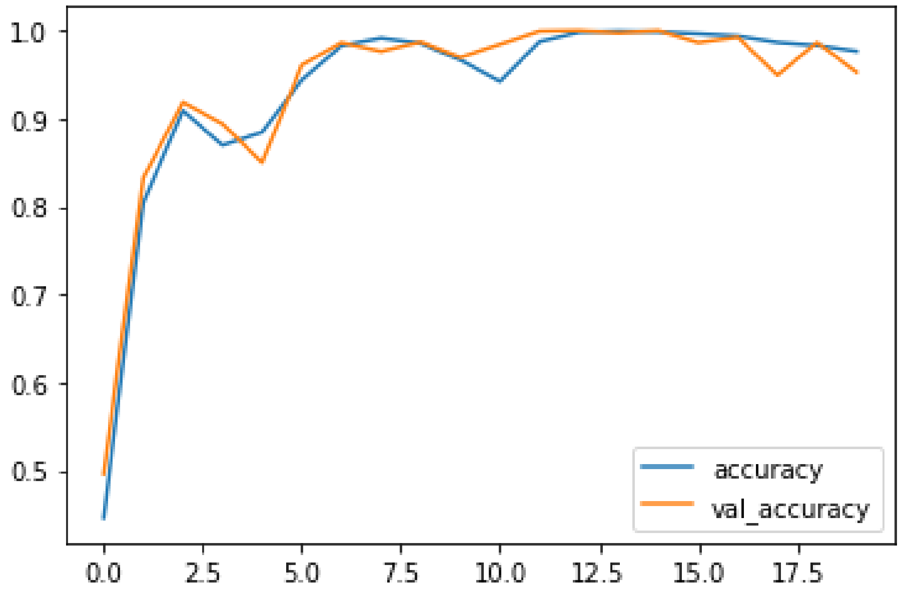

Figure 9: Comparing Accuarcy And Val Accuracy for exp-8

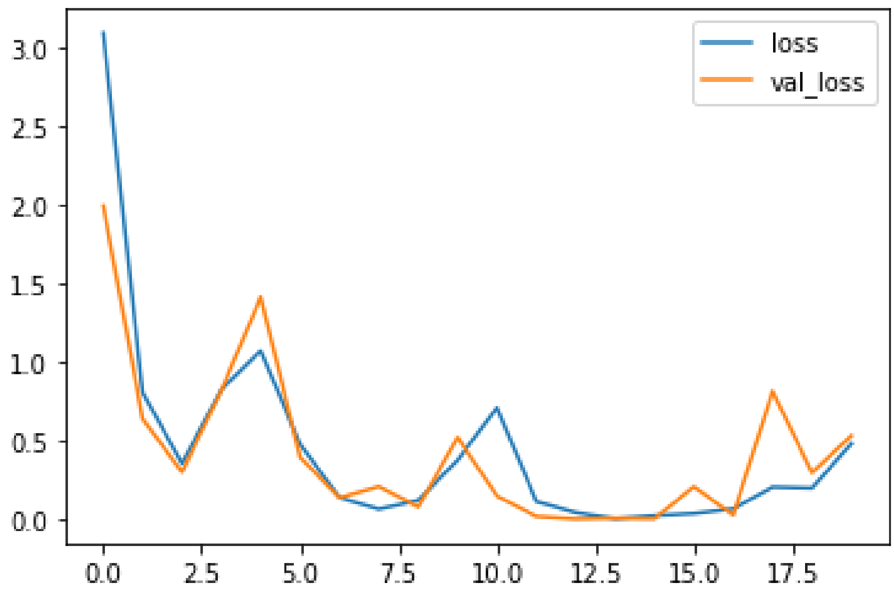

Figure 10: Comparing Loss and Val_Loss for exp-8

The comparison between training and validation accuracy and training and validation loss for experiment 9 is shown in Figure 11 and Figure 12, respectively.

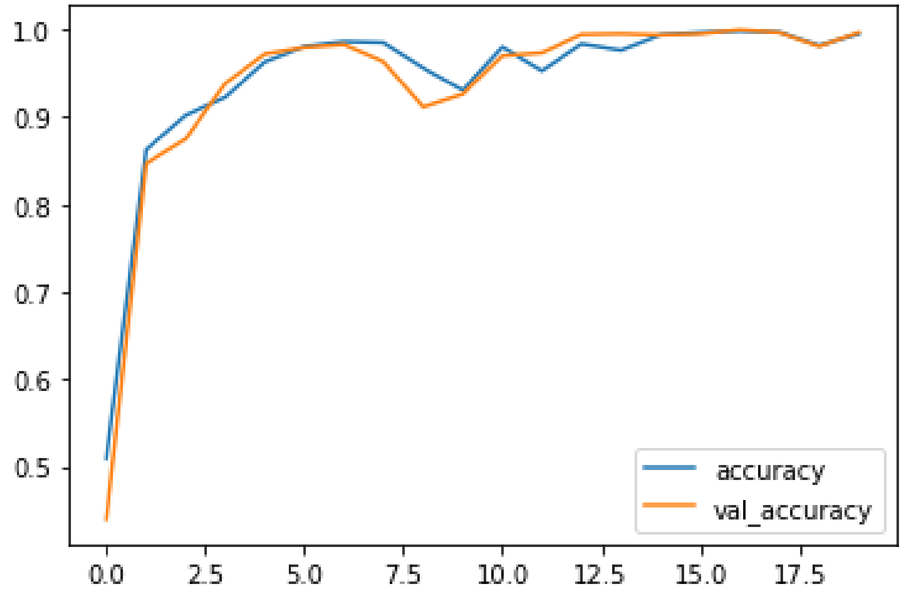

Figure 11: Comparing Accuarcy and Val Accuracy for exp-9

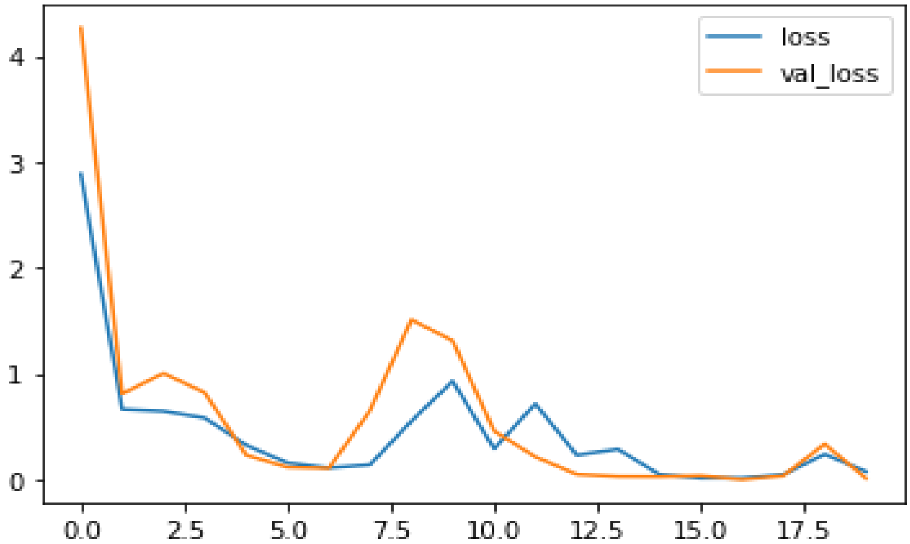

Figure 12: Comparing Loss and Val_Loss for exp-9

The comparison between training and validation accuracy and training and validation loss for experiment 10 is shown in Figure 13 and Figure 14, respectively.

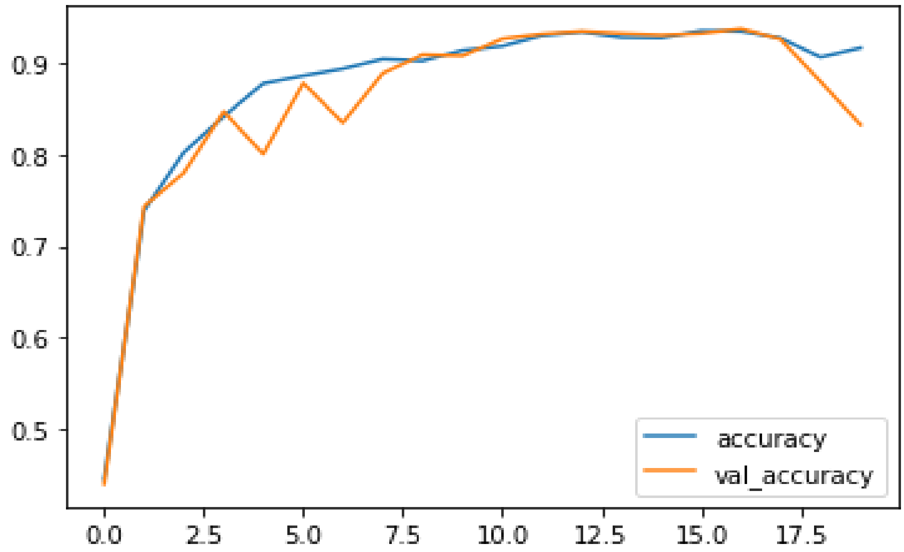

Figure 13: Comparing Accuarcy And Val Accuracy for exp-10

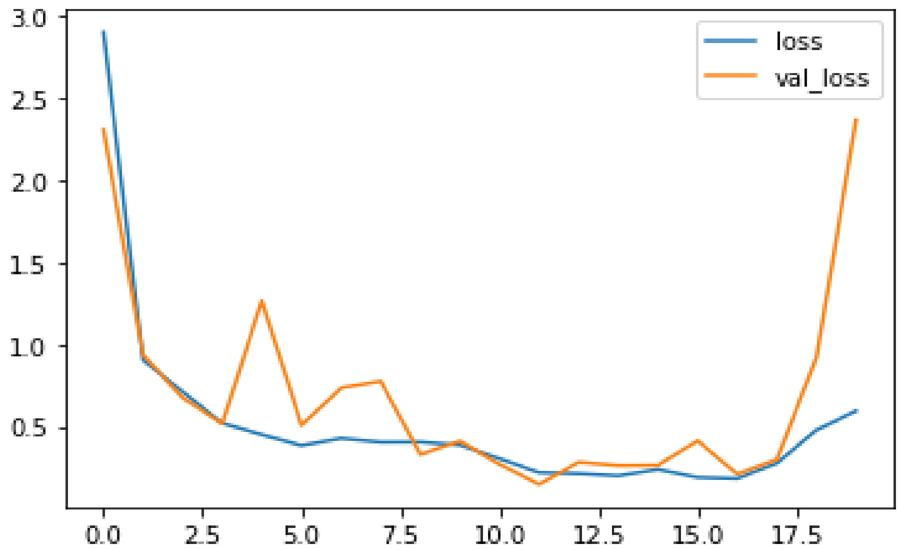

Figure 14: Comparing Loss and Val_Loss for exp_10

Confusion matrix represents the performance of the model on the testing dataset. In the confusion matrix, the column represents the predicted number of samples for a class and rows represent how many actually belong to a class. The confusion matrix for different experiments when evaluated using CNN is shown in Table 6 to Table 11.

Table 6: CONFUSION MATRIX FOR EXPERIMENT 5

		PREDICTED																			
		S01	**S02**	**S03**	**S04**	**S05**	**S06**	**S07**	**S08**	**S09**	**S10**	**S11**	**S12**	**S13**	**S14**	**S15**	**S16**	**S17**	**S18**	**S19**	**S20**
ACTUAL	**S01**	111	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	0	0	9	0
	S02	0	135	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0
	S03	0	0	138	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S04	0	0	0	138	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S05	0	0	0	0	138	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S06	1	4	0	0	0	127	0	0	0	6	0	0	0	0	0	0	0	0	0	0
	S07	4	0	0	0	0	0	130	0	0	0	0	1	3	0	0	0	0	0	0	0
	S08	0	0	0	0	0	0	0	138	0	0	0	0	0	0	0	0	0	0	0	0
	S09	0	0	0	0	0	0	0	0	134	0	0	4	0	0	0	0	0	0	0	0
	S10	0	0	0	0	0	0	0	8	5	124	1	0	0	0	0	0	0	0	0	0
	S11	0	0	0	0	0	0	0	0	0	0	138	0	0	0	0	0	0	0	0	0
	S12	1	0	0	0	0	0	0	0	0	0	0	127	0	0	0	0	0	0	10	0
	S13	0	0	0	0	0	0	0	0	0	0	0	4	134	0	0	0	0	0	0	0
	S14	7	9	0	0	0	0	0	0	0	0	0	0	0	120	0	1	0	1	0	0
	S15	0	0	0	0	6	0	0	0	0	3	0	0	0	0	129	0	0	0	0	0
	S16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	138	0	0	0	0
	S17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	138	0	0	0
	S18	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	134	0	0
	S19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	138	0
	S20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	138

Table 7: CONFUSION MATRIX FOR EXPERIMENT 6

		PREDICTED																			
		S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
ACTUAL	S01	78	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0
	S02	0	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S03	0	0	81	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S04	0	0	0	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S05	0	0	0	0	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S06	0	0	0	0	0	74	0	5	0	0	0	0	0	0	0	4	0	0	0	0
	S07	0	0	0	0	0	0	83	0	0	0	0	0	0	0	0	0	0	0	0	0
	S08	0	0	0	0	0	0	0	83	0	0	0	0	0	0	0	0	0	0	0	0
	S09	0	0	0	14	4	0	0	1	60	1	0	0	0	0	3	0	0	0	0	0
	S10	0	0	0	0	0	0	0	0	0	83	0	0	0	0	0	0	0	0	0	0
	S11	0	0	0	0	0	0	0	0	0	0	83	0	0	0	0	0	0	0	0	0
	S12	0	0	0	0	0	0	0	0	0	0	0	83	0	0	0	0	0	0	0	0
	S13	0	0	0	0	0	0	0	0	0	0	0	0	83	0	0	0	0	0	0	0
	S14	0	0	0	0	0	0	0	0	0	0	0	0	0	83	0	0	0	0	0	0
	S15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	83	0	0	0	0	0
	S16	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	77	0	0	3	0
	S17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	83	0	0	0
	S18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	83	0	0
	S19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	83	0
	S20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	83

Table 8: CONFUSION MATRIX FOR EXPERIMENT 7

		PREDICTED																			
		S01	**S02**	**S03**	**S04**	**S05**	**S06**	**S07**	**S08**	**S09**	**S10**	**S11**	**S12**	**S13**	**S14**	**S15**	**S16**	**S17**	**S18**	**S19**	**S20**
	S01	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S02	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S03	0	0	148	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0
	S04	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S05	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S06	0	0	0	0	0	141	0	0	0	0	0	0	0	0	0	4	0	0	7	0
	S07	0	0	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0
	S08	0	0	0	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0
	S09	0	0	0	0	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0
ACTUAL	**S10**	0	0	0	0	0	0	0	3	0	149	0	0	0	0	0	0	0	0	0	0
	S11	0	3	0	0	0	0	0	0	0	0	149	0	0	0	0	0	0	0	0	0
	S12	0	0	0	0	0	0	0	0	0	0	0	152	0	0	0	0	0	0	0	0
	S13	0	0	0	0	0	0	0	0	0	0	0	0	152	0	0	0	0	0	0	0
	S14	0	0	0	0	0	0	0	0	0	0	0	0	0	152	0	0	0	0	0	0
	S15	0	0	0	0	0	0	0	6	0	0	0	0	0	0	146	0	0	0	0	0
	S16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	152	0	0	0	0
	S17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	152	0	0	0
	S18	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	148	0	0
	S19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	152	0
	S20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	152

Table 9: CONFUSION MATRIX FOR EXPERIMENT 8

		PREDICTED																			
		S01	**S02**	**S03**	**S04**	**S05**	**S06**	**S07**	**S08**	**S09**	**S10**	**S11**	**S12**	**S13**	**S14**	**S15**	**S16**	**S17**	**S18**	**S19**	**S20**
ACTUAL	**S01**	75	0	0	0	0	0	0	0	0	5	0	0	0	0	0	3	0	0	0	0
	S02	0	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S03	0	0	80	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0
	S04	0	0	0	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S05	0	0	0	0	77	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0
	S06	0	1	0	0	5	72	0	0	0	0	0	0	0	0	0	5	0	0	0	0
	S07	0	0	0	0	0	0	81	0	0	0	1	0	0	0	0	1	0	0	0	0
	S08	0	0	0	0	0	0	0	83	0	0	0	0	0	0	0	0	0	0	0	0
	S09	0	0	0	0	0	0	0	0	83	0	0	0	0	0	0	0	0	0	0	0
	S10	0	0	0	0	0	0	0	0	0	81	0	0	0	0	0	0	0	2	0	0
	S11	0	0	1	2	0	0	0	0	0	0	75	0	0	0	0	0	2	0	0	3
	S12	0	0	0	0	0	0	0	0	0	0	0	83	0	0	0	0	0	0	0	0
	S13	0	0	0	0	0	0	0	0	0	0	0	0	81	0	0	2	0	0	0	0
	S14	0	0	0	3	0	0	0	0	0	0	0	0	0	78	0	0	2	0	0	0
	S15	0	8	0	0	0	0	0	1	0	0	0	0	0	0	62	10	0	2	0	0
	S16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	83	0	0	0	0
	S17	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	78	0	0	0
	S18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	83	0	0
	S19	4	5	0	1	0	0	0	1	0	0	0	0	0	0	0	2	0	0	70	0
	S20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	83

Table 10: CONFUSION MATRIX FOR EXPERIMENT 9

		PREDICTED																			
		S01	**S02**	**S03**	**S04**	**S05**	**S06**	**S07**	**S08**	**S09**	**S10**	**S11**	**S12**	**S13**	**S14**	**S15**	**S16**	**S17**	**S18**	**S19**	**S20**
ACTUAL	**S01**	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S02	0	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S03	0	0	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S04	0	0	0	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S05	0	0	0	0	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S06	0	0	0	0	0	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S07	0	0	0	0	0	0	83	0	0	0	0	0	0	0	0	0	0	0	0	0
	S08	0	0	0	0	0	0	0	83	0	0	0	0	0	0	0	0	0	0	0	0
	S09	0	0	0	0	0	0	0	0	83	0	0	0	0	0	0	0	0	0	0	0
	S10	0	0	0	0	0	0	0	0	0	83	0	0	0	0	0	0	0	0	0	0
	S11	0	0	0	0	0	0	0	0	0	0	83	0	0	0	0	0	0	0	0	0
	S12	0	0	0	0	0	0	0	0	0	0	0	83	0	0	0	0	0	0	0	0
	S13	0	0	0	0	0	0	0	0	0	0	0	0	83	0	0	0	0	0	0	0
	S14	0	0	0	0	0	0	0	0	0	0	0	0	0	83	0	0	0	0	0	0
	S15	3	0	0	0	0	0	0	0	0	0	0	0	0	0	80	0	0	0	0	0
	S16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	83	0	0	0	0
	S17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	83	0	0	0
	S18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	83	0	0
	S19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	83	0
	S20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	83

Table 11: CONFUSION MATRIX FOR EXPERIMENT 10

		PREDICTED																			
		S01	**S02**	**S03**	**S04**	**S05**	**S06**	**S07**	**S08**	**S09**	**S10**	**S11**	**S12**	**S13**	**S14**	**S15**	**S16**	**S17**	**S18**	**S19**	**S20**
	S01	89	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0
	S02	8	51	0	0	0	0	0	0	0	1	0	37	0	0	0	0	0	0	0	0
	S03	0	0	97	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S04	0	7	0	80	0	0	0	0	0	3	0	0	0	0	0	7	0	0	0	0
	S05	0	0	0	0	0	92	0	0	2	3	0	0	0	0	0	0	0	0	0	0
	S06	0	0	0	0	0	92	0	0	2	3	0	0	0	0	0	0	0	0	0	0
	S07	1	3	0	0	0	0	77	0	14	2	0	0	0	0	0	0	0	0	0	0
A	**S08**	19	0	0	0	0	0	0	63	0	15	0	0	0	0	0	0	0	0	0	0
C T	**S09**	6	0	0	0	0	0	0	0	90	1	0	0	0	0	0	0	0	0	0	0
U A	**S10**	5	0	0	0	0	0	0	0	0	92	0	0	0	0	0	0	0	0	0	0
L	**S11**	0	0	0	0	0	0	0	0	0	0	89	7	0	1	0	0	0	0	0	0
	S12	9	0	0	0	0	0	0	0	0	3	0	85	0	0	0	0	0	0	0	0
	S13	2	0	0	0	0	0	0	0	0	0	0	0	92	0	0	0	0	0	0	3
	S14	0	4	0	0	0	0	0	0	0	0	0	3	0	90	0	0	0	3	0	0
	S15	2	0	0	0	0	0	0	0	1	2	0	0	0	0	92	0	0	0	0	0
	S16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97	0	0	0	0
	S17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97	0	0	0
	S18	3	0	0	0	0	0	0	0	0	0	6	0	2	0	0	0	0	86	0	0
	S19	0	0	0	0	0	0	3	1	9	3	0	2	0	0	0	0	0	0	79	0
	S20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	95

The training, validation and testing accuracies for different experiments are mentioned in Table 12 below.

Table 12: TRAINING, VALIDATION AND TESTING ACCURACIES

	Average Training Accuracy	Average Validation Accuracy	Average Testing Accuracy
Experiment 05	98.452	97.352	97.424
Experiment 06	97.294	96.248	96.333
Experiment 07	99.266	98.35	98.246
Experiment 08	98.704	97.992	97.826
Experiment 09	98.096	98.678	98.438
Experiment 10	90.574	88.022	88.656

4.2. Results and Parameters

The 7 parameters calculated for user identification are given in the eq (1) to eq (7) as output are as follows:

Sensitivity (RC)

It tells us about how many positive instances of classes are correctly classified [11].

$$\mathbf{Avg\_RC = sum\,(RC)\,/\,row} \quad (1)$$

Accuracy (ACC)

It tells us how many samples are accurately classified [11].

$$\mathbf{Acc = (sum\,(confusion\_matrix.diagonal\,)/\,sum\,(sum\,(confusion\_matrix))\,)} \quad (2)$$

Precision (PR)

It tells us how many are actually positive among the predicted positives [11].

$$\mathbf{Avg\,PR = sum\,((PR))/\,row} \quad (3)$$

F-1 Score (F1)

The harmonic mean of precision and recall is defined as the F-1 Score [11].

$$\mathbf{Avg\ F1\ =\ sum\ (F1)/\ row} \qquad (\mathbf{4})$$

False Acceptance Rate (FAR)

The value representing acceptance of a biometric as from a particular while actually it belongs to some other person [12].

$$\mathbf{Avg\ FAR = sum(FAR)/row} \qquad (\mathbf{5})$$

False Rejection Rate (FRR)

The value representing rejection of a biometric as not from a particular while actually it belongs to that person [12].

$$\mathbf{Avg\ FRR = sum(FRR)/row} \qquad (\mathbf{6})$$

Half Total Error Rate (HTER)

Half total error rate is the half of the sum of False Acceptance Rate and False Rejection Rate [4].

$$\mathbf{Avg\ HTER = (Avg\ FAR + \ Avg\ FRR)/2} \qquad (\mathbf{7})$$

The results of parameters discussed from eq (1) to eq (7) on the testing dataset are listed in Table 13 for identification and in table 14 for authentication.

Table 13: RESULTS FOR IDENTIFICATION

	Exp - 05	**Exp-06**	**Exp-07**	**Exp-08**	**Exp-09**	**Exp-10***
Parameter	Value	Value	Value	Value	Value	Value
Accuracy	0.9590	0.9728	0.9898	0.9481	0.9981	0.8417
Sensitivity (Recall)	0.9590	0.9728	0.9898	0.9481	0.9981	0.8417
Precision	0.9609	0.9751	0.9901	0.9528	0.9982	Nan
F_1 score	0.9588	0.9721	0.9897	0.9478	0.9981	Nan
FAR	0.0010	0.0007	0.0002	0.0013	0.00004	0.0041
FRR	0.0204	0.0135	0.0050	0.0259	0.0009	0.0791
HTER	0.0107	0.0071	0.0026	0.0136	0.0004	0.0416

*- There was some discrepancy in data.

Table 14: RESULTS FOR AUTHENTICATION

	Exp - 05	**Exp-06**	**Exp-07**	**Exp-08**	**Exp-09**	**Exp-10**
Parameter	Value	Value	Value	Value	Value	Value
Accuracy	0.8500	1.0000	0.9500	1.0000	0.8500	0.9500

Chapter 5: Summary, Conclusions and Future Directions

The biometric system using 1-D CNN for identification of different users was developed. The data augmentation and up sampling was performed to deal with the issue of class imbalance and low sample count. After performing the train-test split in a ratio of 0.75:0.25, the train data was further split into validation data and actual training data using the train-test split function in a ratio of 0.7:0.3. The biometric identification system developed leverage the feature extraction ability of 1-D CNN. The average accuracy for authentication across all the experiments is 0.9333 with minimum accuracy of 0.85 for experiment 5 and experiment 9 and maximum accuracy of 1.00 for experiment 6 and experiment 8.

The computational complexity is a criteria which can still be improved for user identification. In future work, implementing LSTM or a combination of LSTM with CNN can be used to further increase the accuracy. Other up-sampling or data augmentation methods can also be applied to increase the size of the database.

References

1. Schons, Thiago, Gladston JP Moreira, Pedro HL Silva, Vitor N. Coelho, and Eduardo JS Luz. "Convolutional network for EEG-based biometric." In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 22nd Iberoamerican Congress, CIARP 2017, Valparaíso, Chile, November 7–10, 2017, Proceedings 22*, pp. 601-608. Springer International Publishing, 2018.

2. Ma, Lan, James W. Minett, Thierry Blu, and William SY Wang. "Resting state EEG-based biometrics for individual identification using convolutional neural networks." In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 2848-2851. IEEE, 2015.

3. Alzahab, N. Abo. "Design and implementation of techniques for the secure authentication of users based on electroencephalogram (EEG) signals." PhD diss., Master's thesis, Marche Polytechnic University, 2021.

4. Marcel, Sebastien, and José del R. Millán. "Person authentication using brainwaves (EEG) and maximum a posteriori model adaptation." *IEEE transactions on pattern analysis and machine intelligence* 29, no. 4 (2007): 743-752.

5. Altahat, Salahiddin Hassan Qodrah. "Robust EEG Channel Set for Biometric Application." PhD diss., University of Canberra, 2017.

6. Sun, Yingnan, Frank P-W. Lo, and Benny Lo. "EEG-based user identification system using 1D-convolutional long short-term memory neural networks." *Expert Systems with Applications* 125 (2019): 259-267.

7. ohamad El-Abed, Christophe Charrier. Evaluation of Biometric Systems. New Trends and Developments in Biometrics, pp. 149 - 169, 2012, ff10.5772/52084ff. ffhal-00990617f

8. Campisi, Patrizio, and Daria La Rocca. "Brain waves for automatic biometric-based user recognition." *IEEE transactions on information forensics and security* 9, no. 5 (2014): 782-800.

9. Ravi, K. V. R., and Ramaswamy Palaniappan. "A minimal channel set for individual identification with EEG biometric using genetic algorithm." In *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, vol. 2, pp. 328-332. IEEE, 2007.

10. Alsumari, Walaa, Muhammad Hussain, Laila Alshehri, and Hatim A. Aboalsamh. "EEG-Based Person Identification and Authentication Using Deep Convolutional Neural Network." *Axioms* 12, no. 1 (2023): 74.

11. Mekruksavanich, Sakorn, and Anuchit Jitpattanakul. "Convolutional neural network and data augmentation for behavioral-based biometric user identification." In *ICT Systems and Sustainability: Proceedings of ICT4SD 2020, Volume 1*, pp. 753-761. Springer Singapore, 2021.

12. Bolle, Ruud M., Jonathan H. Connell, Sharath Pankanti, Nalini K. Ratha, and Andrew W. Senior. *Guide to biometrics*. Springer Science & Business Media, 2013.

13. Abo Alzahab, N., Di Iorio, A., Apollonio, L., Alshalak, M., Gravina, A., Antognoli, L., Baldi, M., Scalise, L., & Alchalabi, B. (2021). Auditory evoked potential EEG-Biometric dataset (version 1.0.0). *PhysioNet*. https://doi.org/10.13026/ps31-fc50.

14. Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., ... & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [Online]. 101 (23), pp. e215–e220.