Contents

[APPENDIX: 3](#_Toc39439829)

[**TASK1** 3](#_Toc39439830)

[1. INTRODUCTION 4](#_Toc39439831)

[2. BACKGROUND 5](#_Toc39439832)

[**2.1 Convolutional Neural Network (CNN)** 5](#_Toc39439833)

[**2.2 Audio Processing** 7](#_Toc39439834)

[**2.2.1 Approaches for audio processing with problems** 7](#_Toc39439835)

[**2.2.2 Sound processing for digit recognition using convolutional neural network** 9](#_Toc39439836)

[**2.2.3 How the data is trained** 9](#_Toc39439837)

[3.Implementation 11](#_Toc39439838)

[**3.1** **Implementation of Convolutional Neural Network** 11](#_Toc39439839)

[**3.2** **Implementation of spoken digit recognition using convolutional neural network** 12](#_Toc39439840)

[4. RESULTS 14](#_Toc39439841)

[5. CONCLUSION 26](#_Toc39439842)

[6. REFERENCES 27](#_Toc39439843)

# APPENDIX:

## **TASK1**

As a part of task one, a poster presentation was demonstrated on the topic of " roulette wheel selection for discrete Particle Swarm Optimization and Tournament selection method to feature selection problem". This task was performed as a group work. The poster created for the task is attached below.

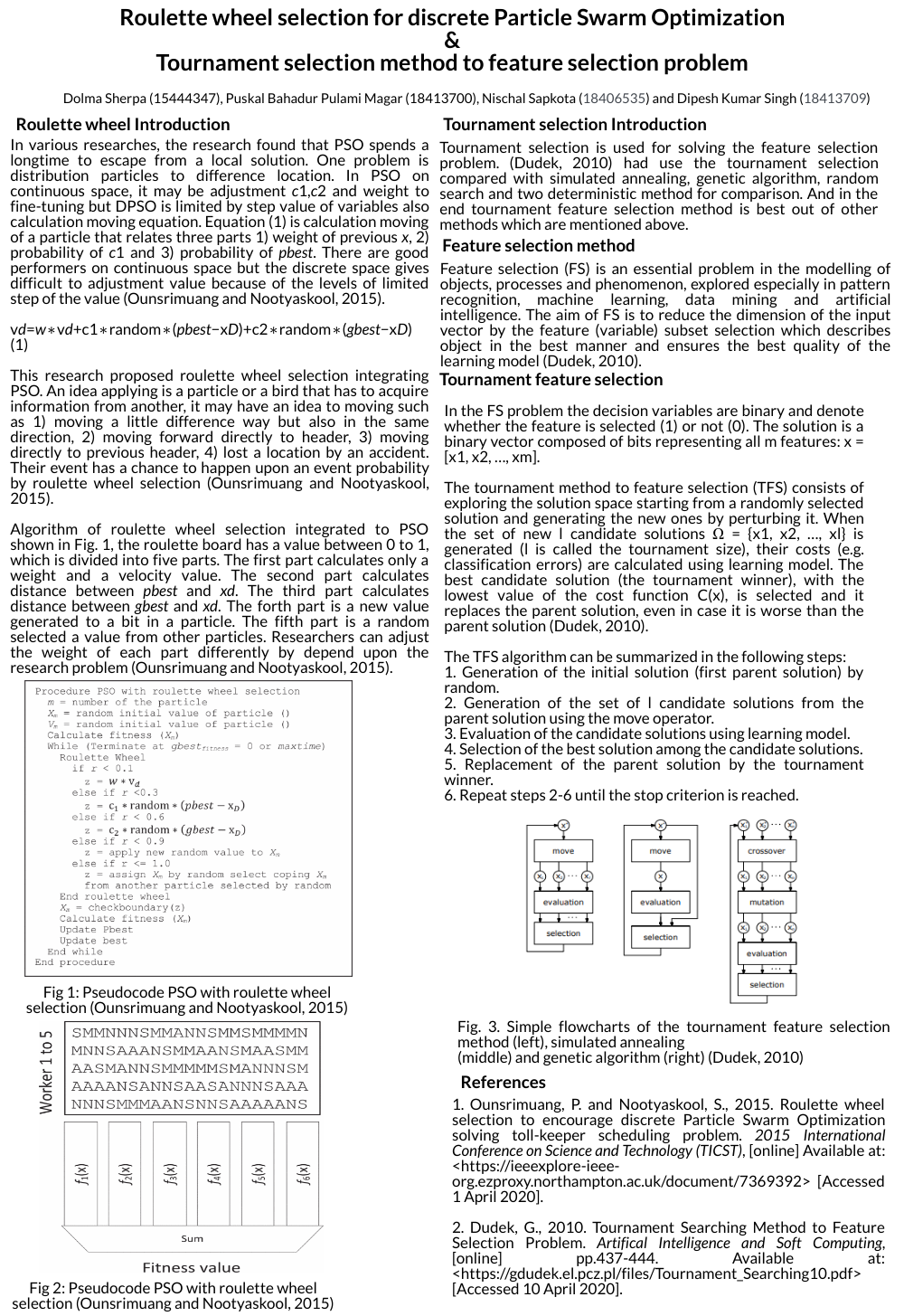


Figure 1: Task 1 (Poster Presentation)

# INTRODUCTION

Spoken digit recognition is an advanced method for digit recognition. This procedure allows the identification of digits through sound i.e. spoken digit. Digits are recognized in the computer field through the specific audio in wav format. It is a process in which detecting a digit in that particular wav and then by using some algorithms to identify which digit is it (*sersc.org,* Test Accuracy Improvement in Spoken Digit Recognition Using Convolutional Neural Networks*,* 2020). Python programming is taken as a benefit as it implements the speech recognition. For the development of speech recognition system, python library offers two methods without using any machine learning methods. The models use huge amount of computing resources and it consumes a lot of time for training the models. There are number of neural network approaches but the convolutional neural network (CNN) is highly preferred for spoken digit recognition.

This paper demonstrates the use of convolutional neural network for detecting spoken digit. It is a feed-forward neural network that extracts the inputted data sets through different convolutional neural networks (CNN). All dataset in CNN is divided into 80% training and 20% testing. For performing this research, the training data set is fed into the system allowing the neural network to learn from inputs. The Spoken Digit Recognition will include data preprocessing stage and model building. There are few spoken digit recognition system that are developed using convolutional neural network (CNN).

The project is based on spoken digit recognition using convolutional neural network in the initial phase. In the second section, it will highlight on how convolutional neural network will be used for audio or sound processing. The third section will demonstrate the implementation of neural network that is followed by various testing approaches. Thus, the final outcome of the system’s solution is explained.

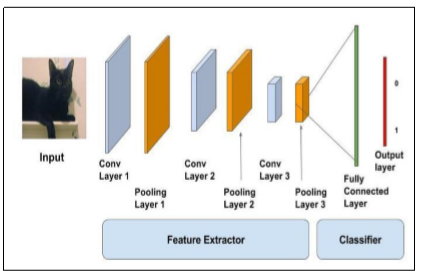
# BACKGROUND

## **2.1 Convolutional Neural Network (CNN)**

Convolutional Neural Network (CNN) is a feed-forward neural network that is used to analyze visual images as it separates spectro-temporal patterns for given input spectrograms. It makes distinctions when sound is masked by time or frequency. The architecture of the CNN comprises of:

1. Convolutional Layer

The Convolutional Layer takes raw audio information and converting it into spectrogram information where each spectogram is taken as an input i.e. picture of the sound. The layer classifies in the same way of image recognition pattern.



**Figure 2: Convolutional Neural Network Architecture (Haoxiang &Lin,2020)**

1. Max Pooling Layer

This layer gradually reduces dimensionality of images and its pixels in the output from previous convolutional layer. It is used for achieving the abstracted form of the representation of the image. Different filters are used to identify different features of image.

1. Dense Layers (MLP)

This layer signifies a matrix vector multiplication by assuming the batch size 1. The values present in the matrix are the trainable parameters which gets updated during backward propagation. Thus, a dense layer is used for changing the vector’s dimension (*Quora.com*, In-Keras-what-is-a-dense-and-a-dropout-layer,2020).

1. Softmax Layer

This layer is an activation function such as sigmoid, tanh and ReLU which is used on the output of the last layer. It is used to determine the output of neural network like yes or no. It maps the resulting values in between 0 to 1 or -1 to 1(*towardsdatascience.com*, activation-functions-neural-networks,2020).

1. Batch Normalization Layer

The layer normalizes the activation of previous layer at each batch that applies transformation to maintain the mean activation close to 0 where the activation deviation is close to 1(*keras,* Normalization Layer*,* 2020). It allows each layer of a network to learn by itself independently compared to other layers.

1. Dropout Layer

In this layer randomly selected neurons are ignored during training. As a result, the contribution to the activation of downstream neurons that is temporarily detached on the forward pass. The updates of the weight are not applied to the neuron on the backward pass(*machinelearningmastery.com*, Dropout regularization-deep-learning-models-keras,2020).

## **2.2 Audio Processing**

Audio processing is the process of listening to and analyzing audio recordings. This method is used mostly used in AI technologies such as virtual assistants, automatic speech recognition and text to speech application (*lionbridge.ai*, what-is-audio-processing,2020).

## **2.2.1 Approaches for audio processing with problems**

There are three different approaches for spoken digit recognition. The approaches are based on simple neural network, spectrogram and **Mel-Frequency Cepstrum Coefficient (MFCC).**

1. **Simple Neural Network Approach:**

**It loads the wav file which is taken as an input from spectrogram. The wav file loads as a NUMPY array and feeds numpy array to a simple multi-layer perceptron. On converting, the wav file to NUMPY array then the data gets stored as a 1-D matrix. But the length of the data is not specific as it highly depends on data. As a result, the first layer of the Multi-layer perceptron will have ~1000 neurons which add extreme complexity for further calculations (Adhishthite, Sound-mnist,2020).**

1. Spectrogram Approach

By converting the wav data into a spectrogram i.e. image file. The image file is fed to Simple neural network with 4096 neurons in the first layer. The spectrogram approach is good but the number of neurons are large as it is not logical for flattening out an image and feeding it to simple neural network.

But if we feed image to a simple convolutional neural network (CNN). Then every audio will be converted into a simple 2-D image, and will be fed to a CNN.As a result, it will speed up the training. Since, CNNs are very good at image recognition it will provide good output (**Adhishthite, Sound-mnist,2020**).



**Figure 3: Spectogram (Adhishthite, Sound-mnist,2020)**

1. Mel- Frequency Cepstrum Coefficient (MFCC) Approach):

MFCC takes the human perception for sensitivity at applicable frequencies by converting   
 the conventional frequency to Mel Scale which is appropriate for speech recognition ([Joyjit Chatterjee](https://www.researchgate.net/profile/Joyjit_Chatterjee),2020).

In sound processing, the mel-frequency cepstrum (MFC) demonstrates the short-term power spectrum of a sound that is based on a linear cosine transforms a log power spectrum on a nonlinear mel scale of frequency. Mel-Frequency Cepstrum Coefficients (MFCCs) are coefficients which collectively makes up an MFC. It is derived from a type of cepstral representation of the audio clip (a nonlinear “spectrum-of-a-spectrum”). There is difference between the cepstrum and mel-frequency cepstrum. In the MFC, the frequency bands are equally spaced on the mel scale which approximates the human auditory systems response more closely than the linearly-spaced frequency bands used in normal cepstrum. This frequency warping can permit better illustration of audio example in audio compression (**Adhishthite, Sound-mnist,2020**).

MFCC is better illustration of sound which can be treated like an image for all practical purposes for training. Thus, MFCC approach is best for WAV files. Therefore, this system includes MFCC approach (**Adhishthite, Sound-mnist,2020**).

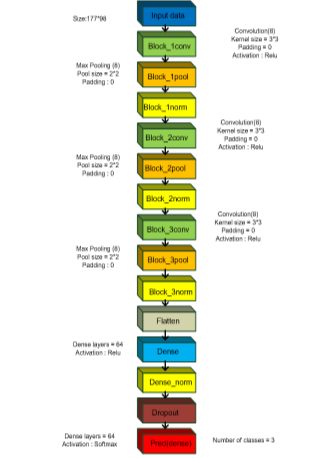
### **2.2.2 Sound processing for digit recognition using convolutional neural network**

In Sound processing for digit recognition, the WAV file is fed into the convolutional layer. Then every audio will be converted into a simple 2-D image, and will be fed to a CNN.As a result, it will speed up the training (**Adhishthite, Sound-mnist,2020**).

In Convolutional Neural Network the entire data is divided into training (80%) and testing (20%).At first the data preprocessing is included for extracting data from WAV files. Then the data is passed into convolutional layer along with max pooling layer by extracting the data of WAV files. Then, it is passed on to MFCC for speech recognition based on appropriate frequency. Thus, it is then conveyed to full connected layer.

### **2.2.3 How the data is trained**

If the convolutional neural network has four layers C1, C2, C3 and output layer. There are various number of feature maps in every layer altogether there are thirty-two feature map. The subjects are trained using convolutional neural network (CNN). The outcome gained from the training is good with more accuracy.



**Figure 4: CNN architecture**

There are different parameters for training which can help in improving accuracy and time taken for training. Some of the parameters for training are:

1. Batch Size

It is a number of datasets given to a network per iteration as the number of batch sizes affect the model training time. If a complete dataset is passed through the network on each iteration, then the training time maybe less (Dey & Learning,2020).

1. Epoch

It is the number of times the dataset is being given completely to the network. For example, if the dataset size is 1100 and the batch size is 100 it would require 11 iterations to complete one epoch. Epochs helps in reusing the same datasets for training repeatedly(WAZIR,2020).

1. Loss

It is the value that is calculated after each iteration to define the error. For instance, if the dataset of a number is 0-9 and the given dataset for training is 9, the output of the training will produce probabilities for each class of numbers. There are different loss functions in deep learning such as Cross-entropy for softmax layers used in the system that is used in back-propagation error. As a result, the error becomes the difference of the prediction and target (Ankit et al.,2020).

1. Optimizer

Optimizer benefits by reducing the output error of the loss function on the basis of change in weights and bias values in the model.

1. Activation Function

There is various activation function used in deep learning such as sigmoid, Softsign and Softmax. The main use of activation function is to choose whether neuron should fire data or not. It is triggered by obtaining the value received from the neuron reevaluating it (Deng & Platt,2020).

# 3.Implementation

In training large amount of data, we need a lot of resources in each layer based on different python libraries.

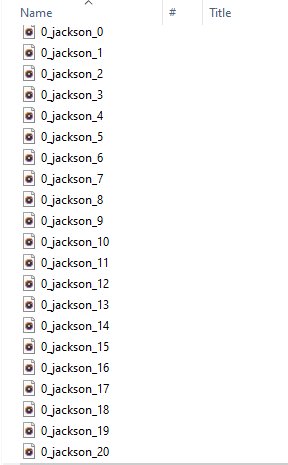
### **Implementation of Convolutional Neural Network**

Convolution Neural Network is used in Sound processing application as all the images are taken directly by the model to process. The data samples in CNN are also the same but there are different features. The features are extracted from the spectrograms and later with the use of extracted features. In the initial layers, the filters are passing like horizontal, vertical and diagonal lines to induce only a map of the image edges. It is easy to train and have fewer parameters in comparison to fully-connected networks with the same number of hidden units (Haoxiang & Lin, 2020).

The data illustrates the samples given to the feature extractor. After that the extracted features are taken as an input parameters and fed as an input into the Keras model. The feature extraction is then followed by max pooling layer and normalization layer with 3 dense layers (MLP), Softmax Activation for output ,BatchNormalization Layer after every Conv Layer and Dense Layer and Dropout for every layer of MLP (**Adhishthite, Sound-mnist,2020**).

## **Implementation of spoken digit recognition using convolutional neural network**

This part describes about the spoken digit recognition system using sound recognition library. The data set are used in audio format (WAV).

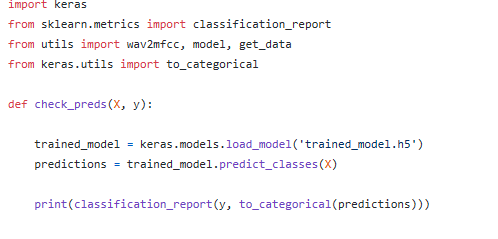


**Figure 5: Dataset for Spoken Digit recognition**

****

**Figure 6: Training the network**

This section of code is executed for training the networks against the data set obtained by learning different frequencies and amplitude of sound.



**Figure 6: Testing the network**

The code permits the user to use the training model for recognizing the spoken digits.

# RESULTS

When the implementation phase of the system is completed then the testing is performed in the basis of the output obtained.

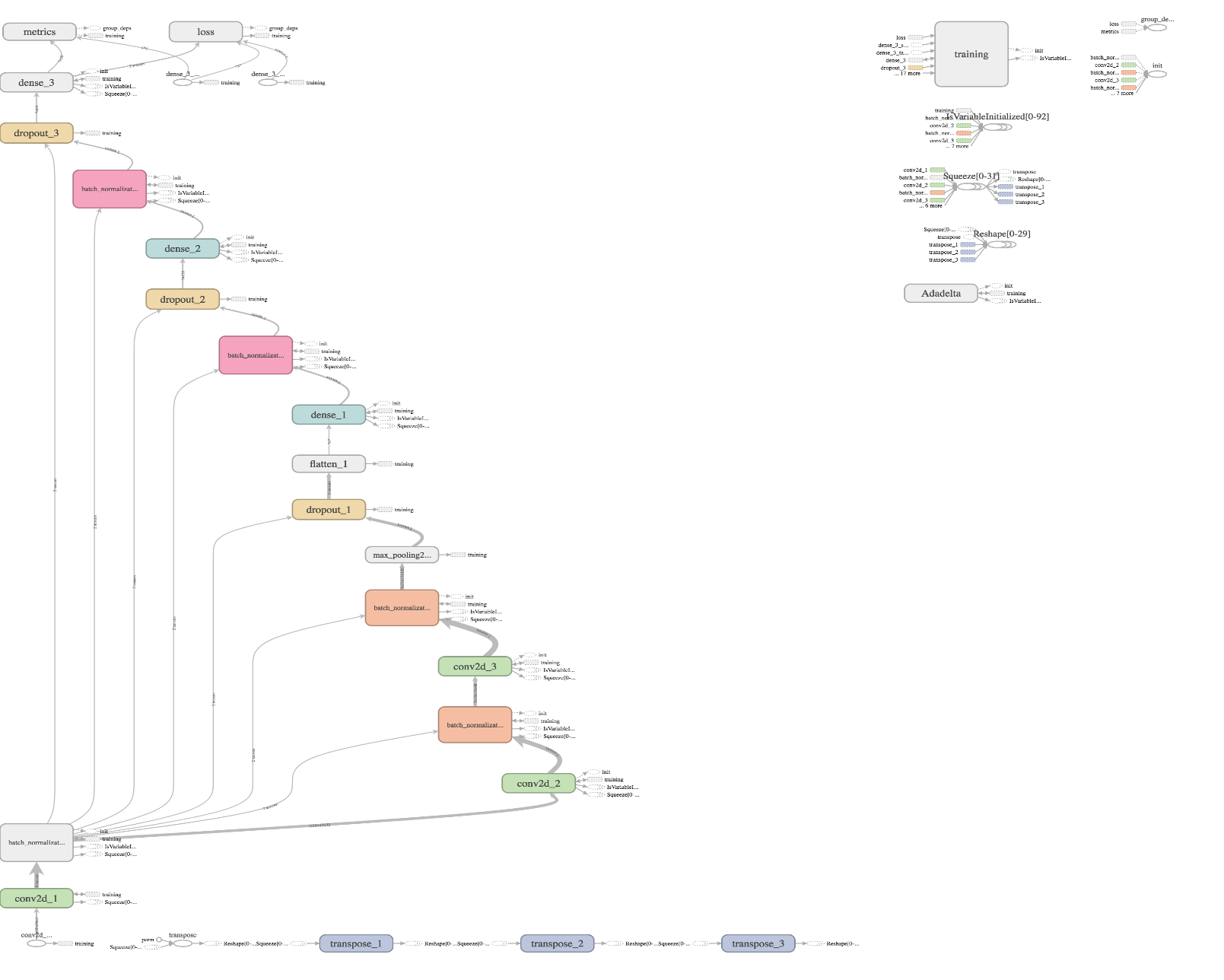
The model of the system is build using keras. The models are as follows:

1. **Model Hyper parameters**
   1. Optimizer - Adadelta
   2. Activation - ReLU
   3. Number of epochs - 50
   4. Batch Size - 64
   5. Learning rate - Adadelta default
   6. Loss - Categorical Crossentropy
2. **Model Structure**
   1. 3 convolutional layers
   2. 1 Max Pooling Layer
   3. 3 dense layers (MLP)
   4. Softmax Activation for output
   5. Batch Normalization Layer after every ConV Layer and Dense Layer.
   6. Dropout for every layer of MLP.

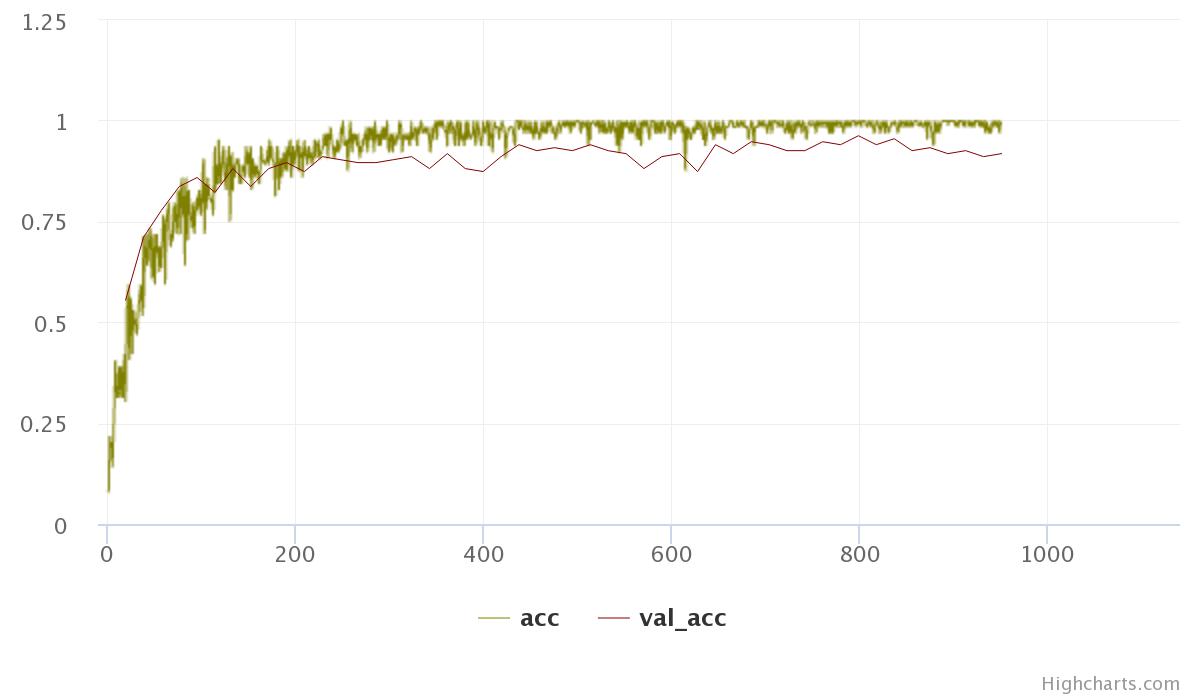
## **Output testing and evaluation**

|  |  |
| --- | --- |
| Input | Output |
| Running file main.py  py main.py |  |
| Running instances of experiment while running main.py file using comet |  |
| Running instances of experiment while running main.py file using comet and comparison of different parameters |  |
|  |  |
| Running file main.py with output |

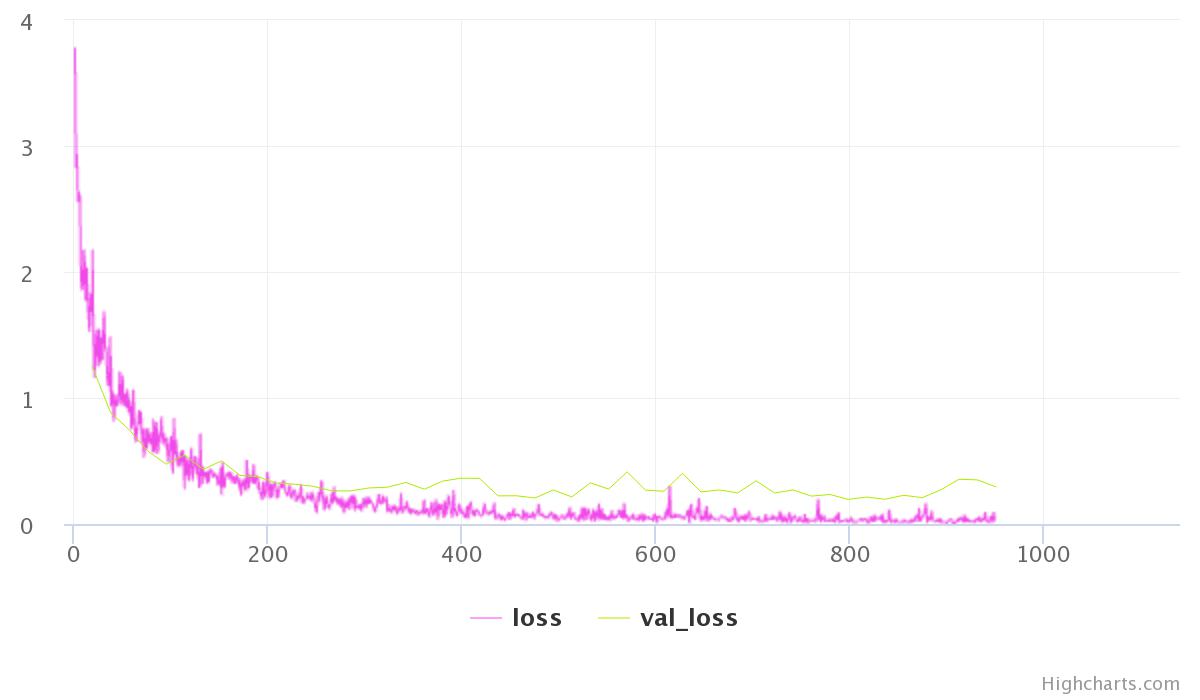
|  |  |
| --- | --- |
| Running hyper parameters on the basis of keras model |  |
| Running metrics |  |
| Output |  |
| Histograms |  |
| Charts |  |



**Figure 8: Tensorboard Visualization of Model**

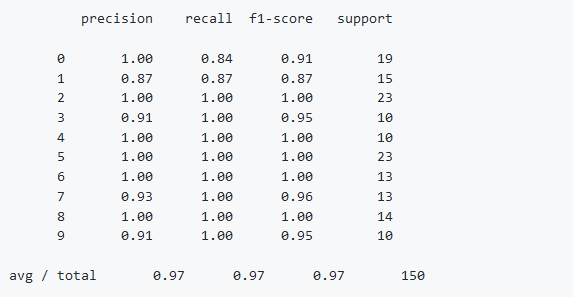
****

**Figure 9: Model Accuracy graph**

****

**Figure 10: Model Loss graph**

The gained accuracy from the model is 98%.

****

**Figure 20: Model Performance**

After running the system, with the help of various libraries and use of keras-tensorflow backend to generate the tests. It was found successful. The test shows the 97% accuracy. As a result, Neural Network is trained properly to classify spoken digits.

# CONCLUSION

The assessment helps in evaluating the use of convolutional neural network(CNN) for spoken digit recognition. CNN (convolutional neural network) is mostly used for image recognition but in sound processing it converts the spectrogram into image and takes it as an input. On the basis of training and frequency of the sound the data is categorized. The system uses different tools and libraries for spoken digit recognition. The output obtained from various tools shows the loss and accuracy of batch on the basis of keras mode building.

# REFERENCES

1. Test Accuracy Improvement in Spoken Digit Recognition Using Convolutional Neural Networks | International Journal of Advanced Science and Technology . 2020. Test Accuracy Improvement in Spoken Digit Recognition Using Convolutional Neural Networks | International Journal of Advanced Science and Technology . [ONLINE] Available at: http://sersc.org/journals/index.php/IJAST/article/view/3387. [Accessed 01 May 2020].
2. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/CNN\_ASLPTrans2-14
3. AI Graduate Admin. 2020. Audio Classification Using CNN — An Experiment - AI Graduate - Medium. [ONLINE] Available at: https://medium.com/x8-the-ai-community/audio-classification-using-cnn-coding-example-f9cbd272269e. [Accessed 01 May 2020].
4. Robust Recognition of Tone Specified Mizo Digits Using CNN-LSTM and Nonlinear Spectral Resolution - IEEE Conference Publication. 2020. Robust Recognition of Tone Specified Mizo Digits Using CNN-LSTM and Nonlinear Spectral Resolution - IEEE Conference Publication. [ONLINE] Available at: https://ieeexplore.ieee.org/document/8986974. [Accessed 01 May 2020].
5. Arthur Juliani. 2020. Recognizing Sounds (A Deep Learning Case Study) - Arthur Juliani - Medium. [ONLINE] Available at: https://medium.com/@awjuliani/recognizing-sounds-a-deep-learning-case-study-1bc37444d44d#.5ubhfdh0h. [Accessed 01 May 2020].
6. CS231n Convolutional Neural Networks for Visual Recognition. 2020. CS231n Convolutional Neural Networks for Visual Recognition. [ONLINE] Available at: https://cs231n.github.io/convolutional-networks/#pool. [Accessed 01 May 2020].
7. F D. 2020. Batch normalization in Neural Networks - Towards Data Science. [ONLINE] Available at: https://towardsdatascience.com/batch-normalization-in-neural-networks-1ac91516821c. [Accessed 02 May 2020].

8) Normalization Layers - Keras Documentation. 2020. Normalization Layers - Keras Documentation. [ONLINE] Available at: https://keras.io/layers/normalization/. [Accessed 02 May 2020].

9) Machine Learning Mastery. 2020. Dropout Regularization in Deep Learning Models With Keras. [ONLINE] Available at: https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/. [Accessed 02 May 2020].

10) AI Graduate Admin. 2020. Audio Classification Using CNN — An Experiment - AI Graduate - Medium. [ONLINE] Available at: https://medium.com/x8-the-ai-community/audio-classification-using-cnn-coding-example-f9cbd272269e. [Accessed 02 May 2020].

11) SAGAR SHARMA. 2020. Activation Functions in Neural Networks - Towards Data Science. [ONLINE] Available at: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6. [Accessed 02 May 2020].

12) GitHub. 2020. GitHub - adhishthite/sound-mnist: A Convolutional Neural Network to identify spoken digits. [ONLINE] Available at: <https://github.com/adhishthite/sound-mnist>. [Accessed 01 May 2020].

13) Sādhanā. 2020. Enhancing spoken connected-digit recognition accuracy by error correction codes — A novel scheme | SpringerLink. [ONLINE] Available at: https://link.springer.com/article/10.1007/BF02703262. [Accessed 02 May 2020].

14) ResearchGate. 2020. (PDF) Models of spoken-word recognition. [ONLINE] Available at: https://www.researchgate.net/publication/264206266\_Models\_of\_spoken-word\_recognition. [Accessed 02 May 2020].

15) Spoken Word Recognition - an overview | ScienceDirect Topics. 2020. Spoken Word Recognition - an overview | ScienceDirect Topics. [ONLINE] Available at: <https://www.sciencedirect.com/topics/psychology/spoken-word-recognition>. [Accessed 02 May 2020].