A PRESENTATION ON 'NEPALI BARNA RECOGNITION'

PREPARED BY:

Ronish Magar (43788)

Roshan Bhattarai (43790)

Rupesh Thakur (43792)

Saroj Subedi (43797)

SUPERVISED BY:

Shyam Krishna Khadka

(Nepal Telecom,

Lecturer,

Himalaya College of Engineering)

2018-08-11

INTRODUCTION:

- **Nepali Barna Recognition**
  - Speech Recognition System
  - Predicts the given speech input as text output.
- **Speech Recognition:**
  - Decoding acoustic speech signal captured by microphone or telephone, to a set of words.
  - Also known as "automatic speech recognition" (ASR), "computer speech recognition", or just "speech to text" (STT)

PROBLEM STATEMENT:

- Less works in the development of Nepali ASR (Automatic Speech Recognition).
- More works done in English ASR.
- Unavailability of Virtual Assistants in Nepali.

OBJECTIVES:

- To develop Convolutional Neural Network (CNN) based model for Nepali Speech Recognition.
- To learn more about various aspects of neural networks.

SCOPES AND APPLICATION:

- Assistive applications for disabled people.
- Controlling voice-controlled equipment.
- Base for virtual assistant applications in Nepali.
- Educational Software,

METHODOLOGY

SYSTEM OVERVIEW

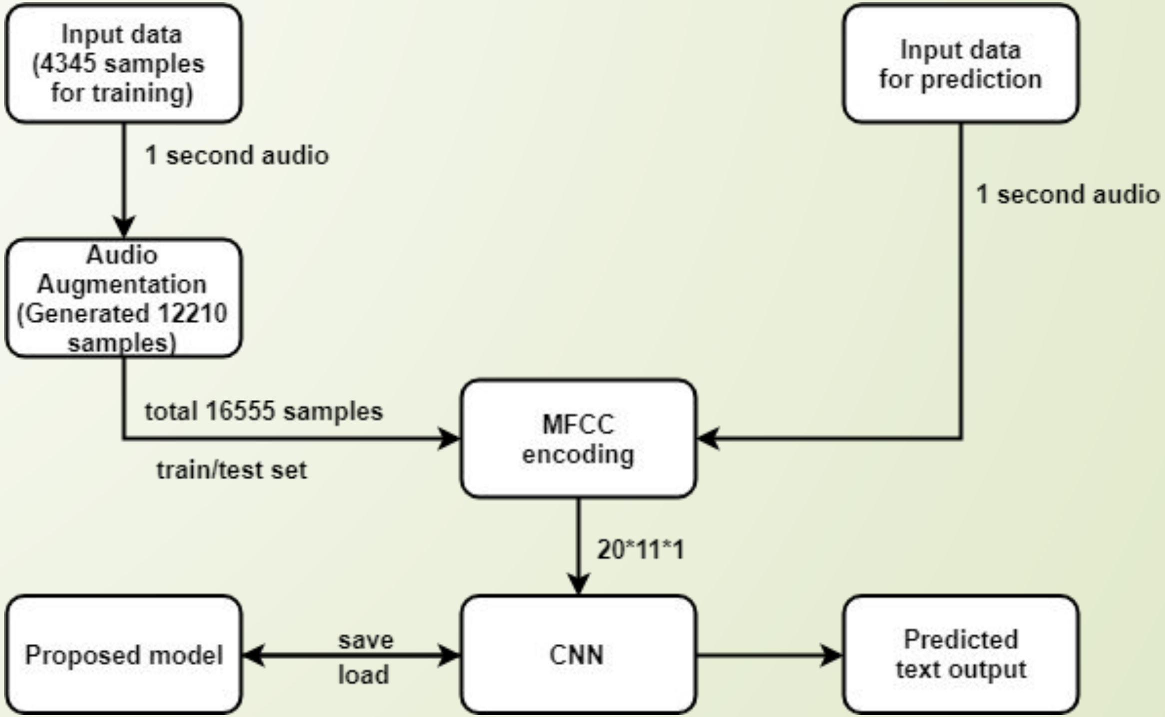


Fig: System Block Diagram

2018-08-11

INPUT DATA:

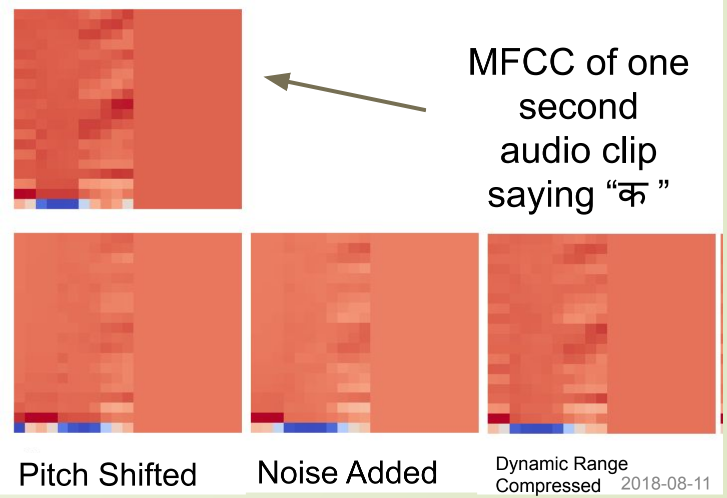
- **Creating Data-sets:**
  - Recording 1 second mono (through one channel) audio for every Nepali alphabets
  - Labeling them in each folder
  - A total of 4345 audio samples were recorded.
- **Prediction:**
  - Similarly, for prediction 1 second mono audio is recorded from the web app.

AUDIO AUGMENTATION:

- Common strategy adopted to increase the quantity of training data.
- It adds some random but nearest values to standard data.
- Different techniques used for data Augmentation:
  - Pitch Shifting
  - Dynamic Range Compression
  - Noise addition

´12210 more data were

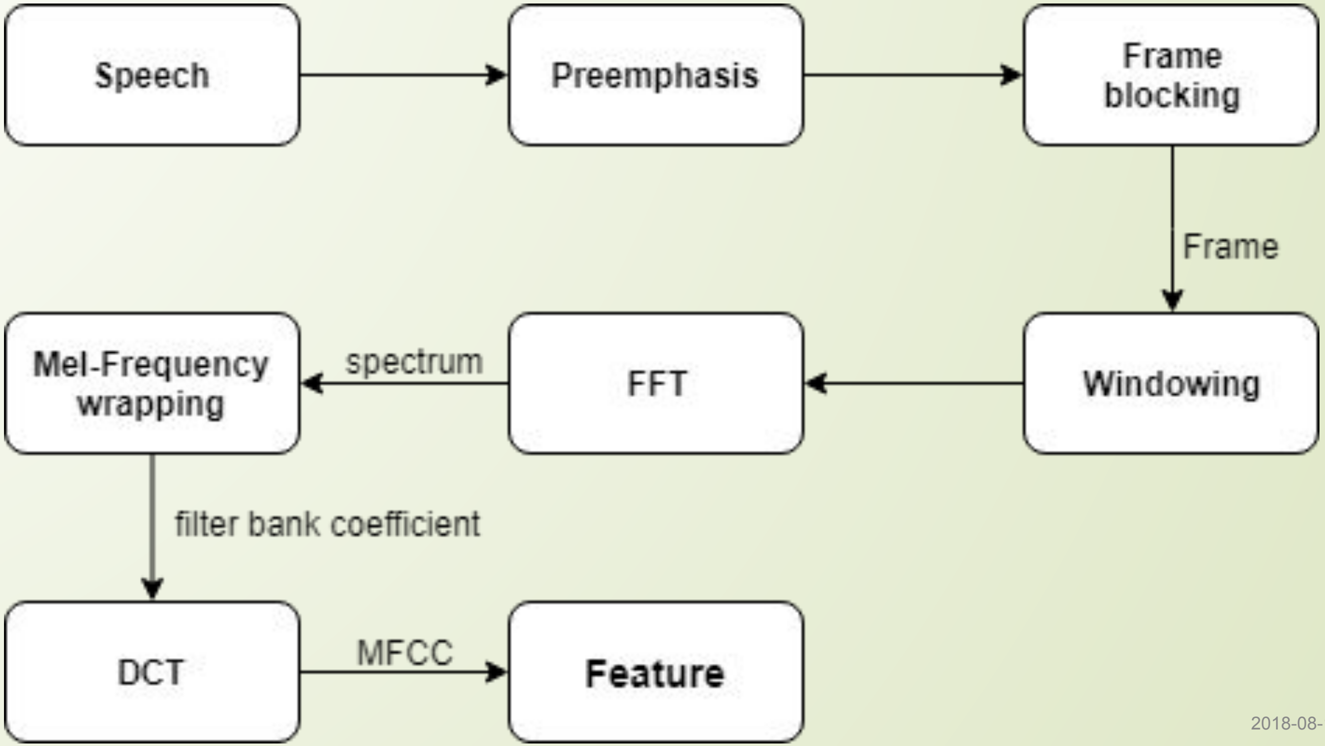
created by data augmentation



MFCC: Mel Frequency Cepstral Coefficients

- Feature extraction techniques used in speech recognition.
- Audio files can’t be directly fed to convolutional network.
- MFCC encoding actually converts the audio into sort of image like data.
- Audio files are encoded as vectors.
- Fixed size vector are created for each audio files.
- MFCC contains the energy/intensity values of a pixel.

MFCC



Speech

Preemphasis

Frame blocking

Frame

Windowing

FFT

spectrum

Mel-Frequency wrapping

filter bank coefficient

DCT

MFCC

Feature

ALGORITHM FOR MFCC:

Step 1: Get the audio signal

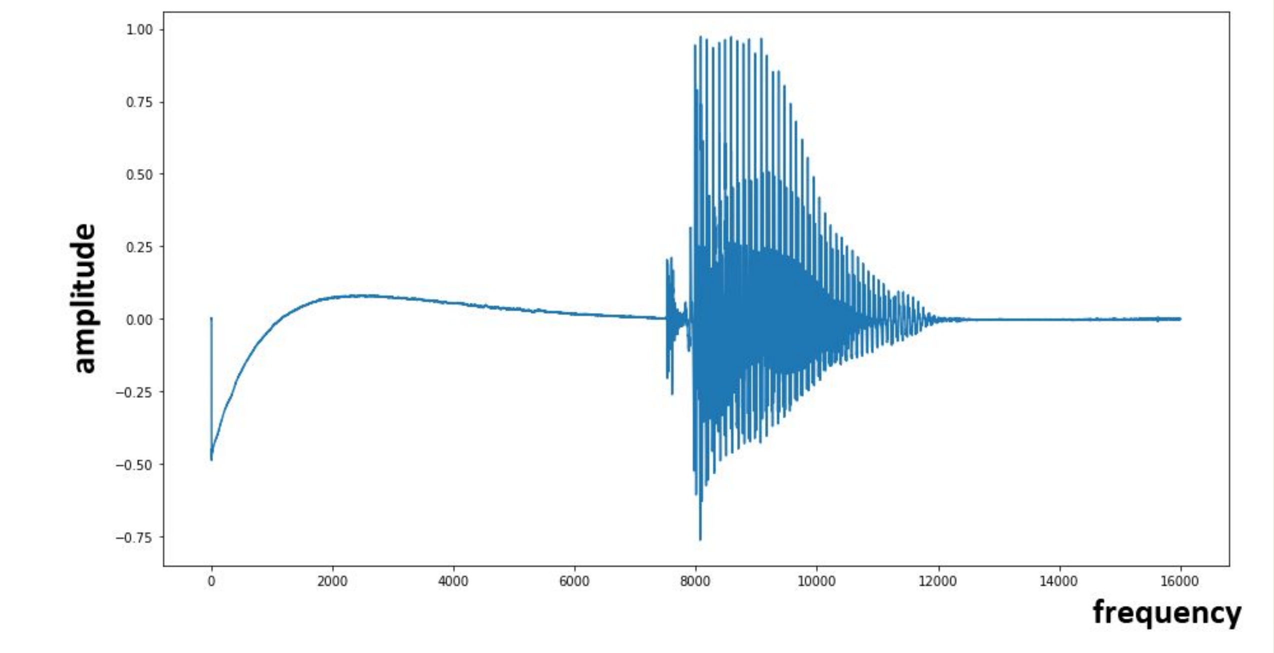


Fig: Applying FFT on the window

Step 2: Pre emphasis Filter

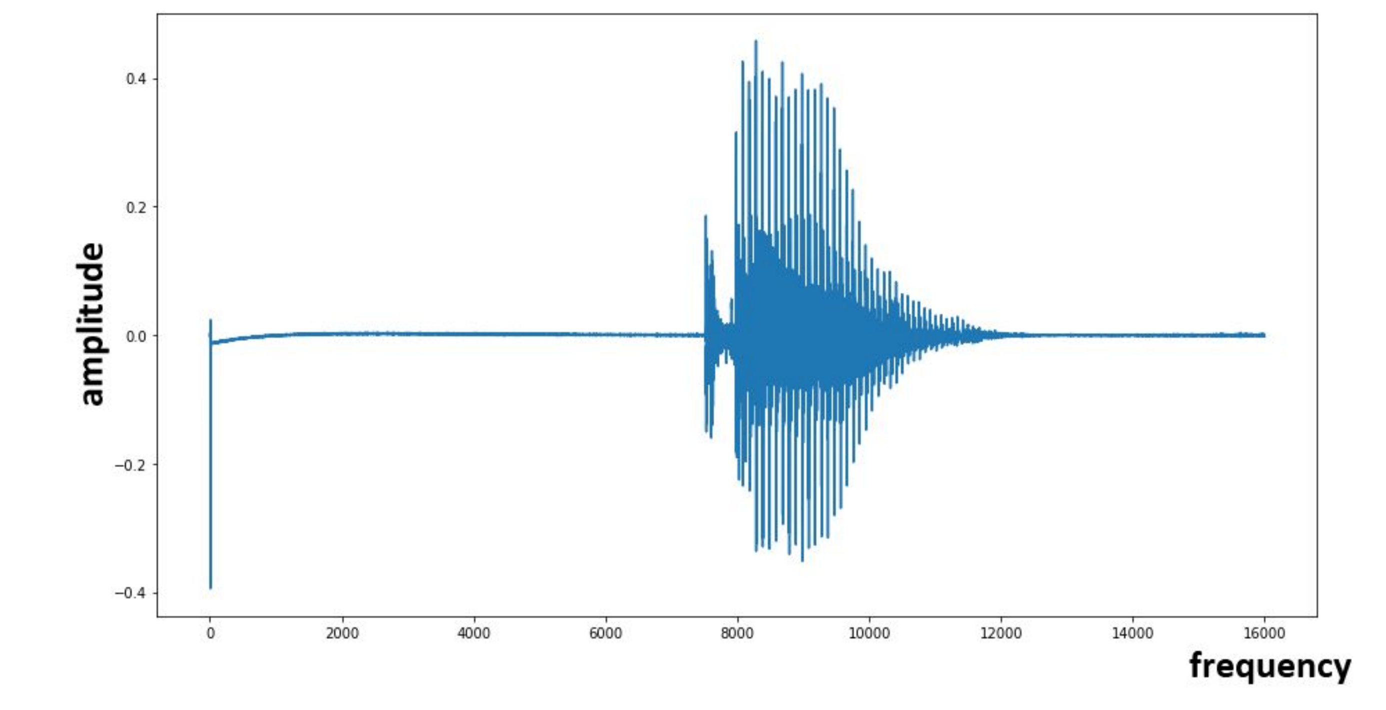


Fig: Applying FFT on the window

Step 3: Framing

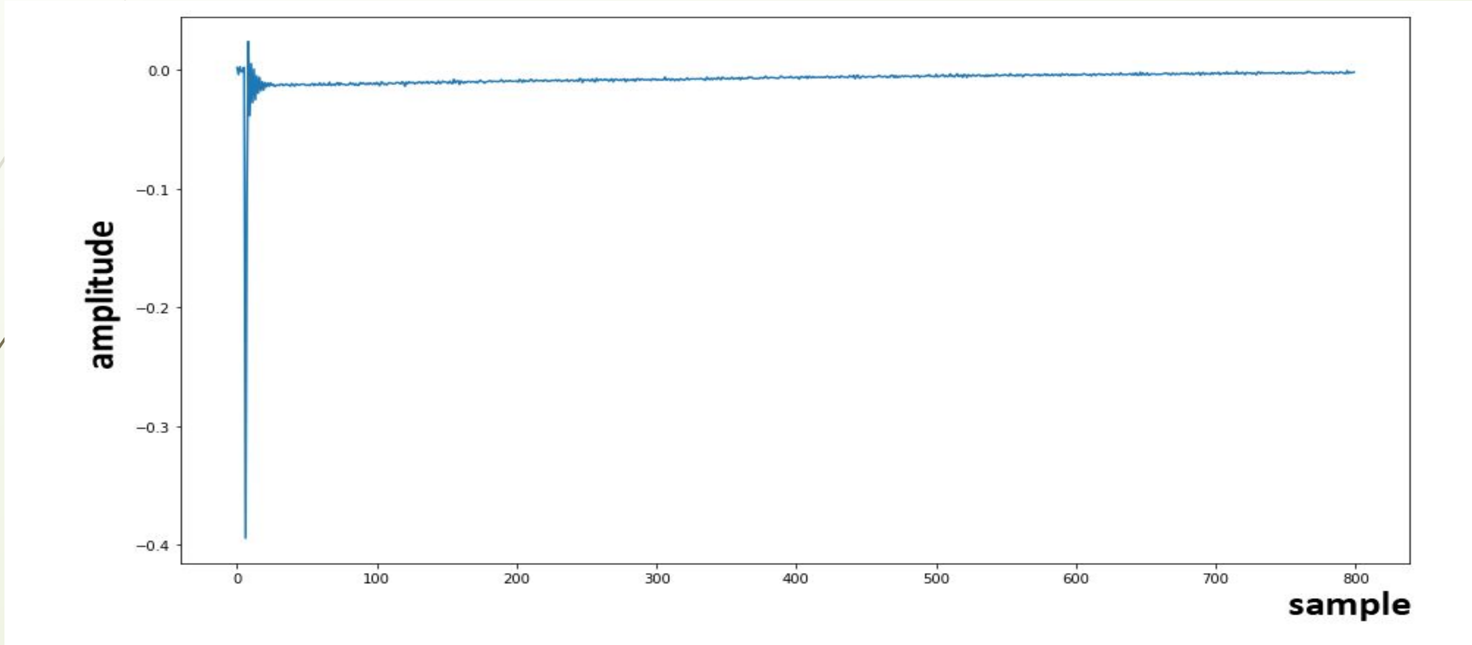
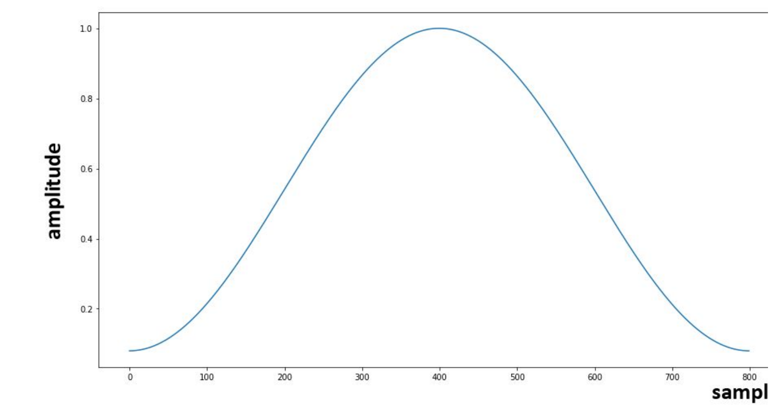


Fig: First frame with 800 samples

Step 4:Windowing



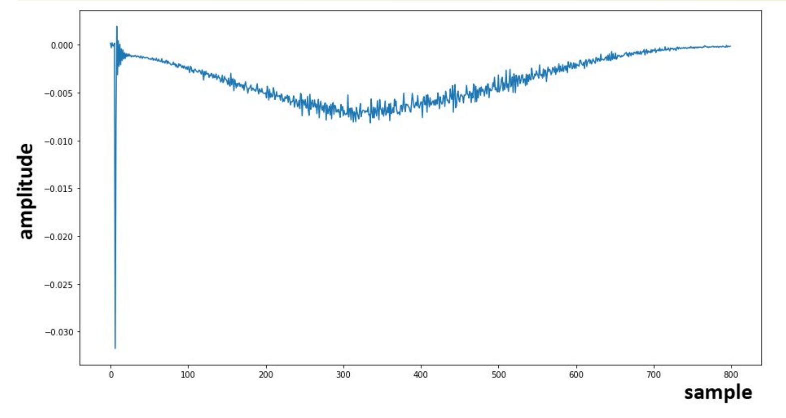


Fig: Hamming window with 800 samples

Fig: Hammed frame

Step 5: Fast Fourier transform

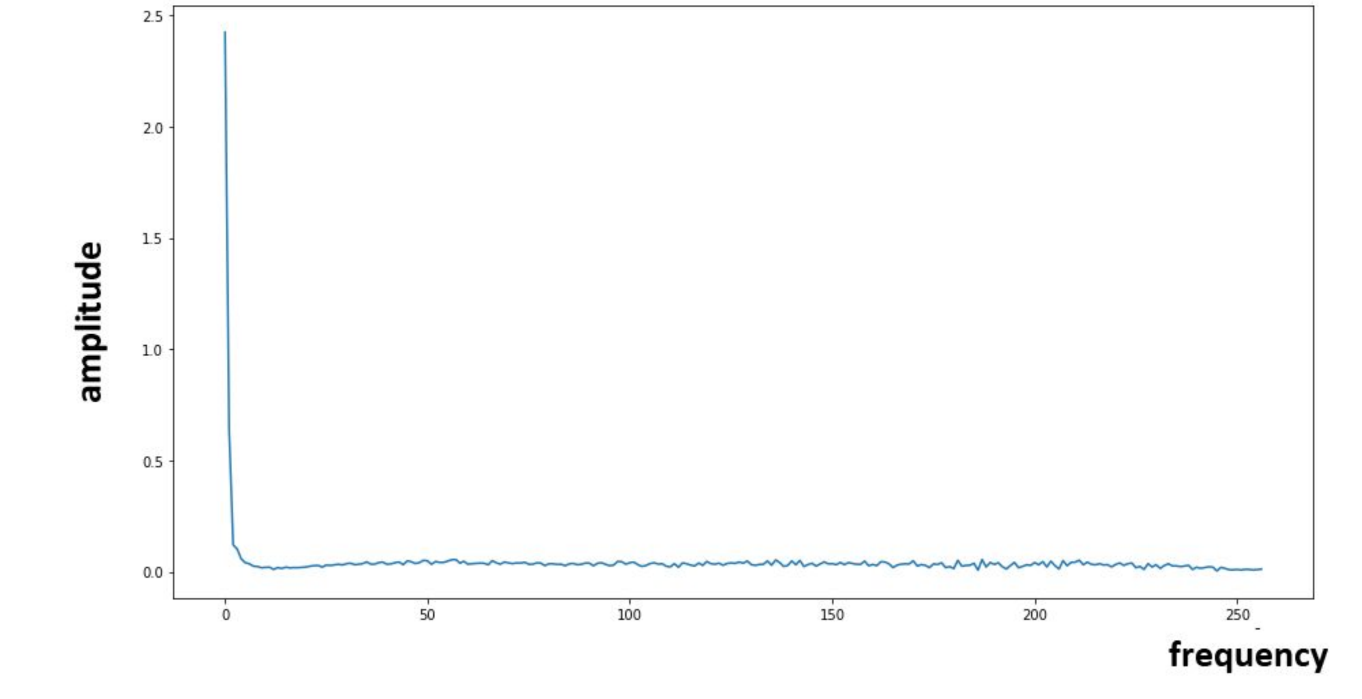


Fig: Applying FFT on the window

Step 6: Generating power spectrum

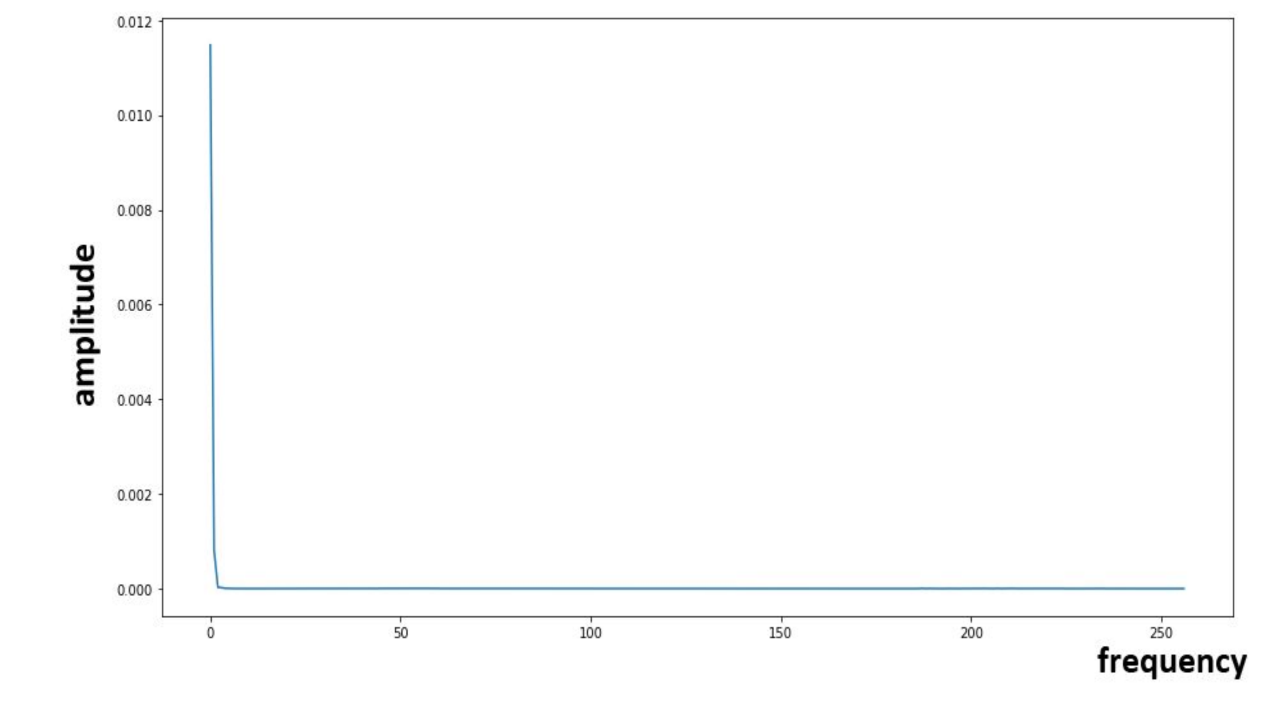


Fig: Applying FFT on the window

Step 7: Mel filter bank processing

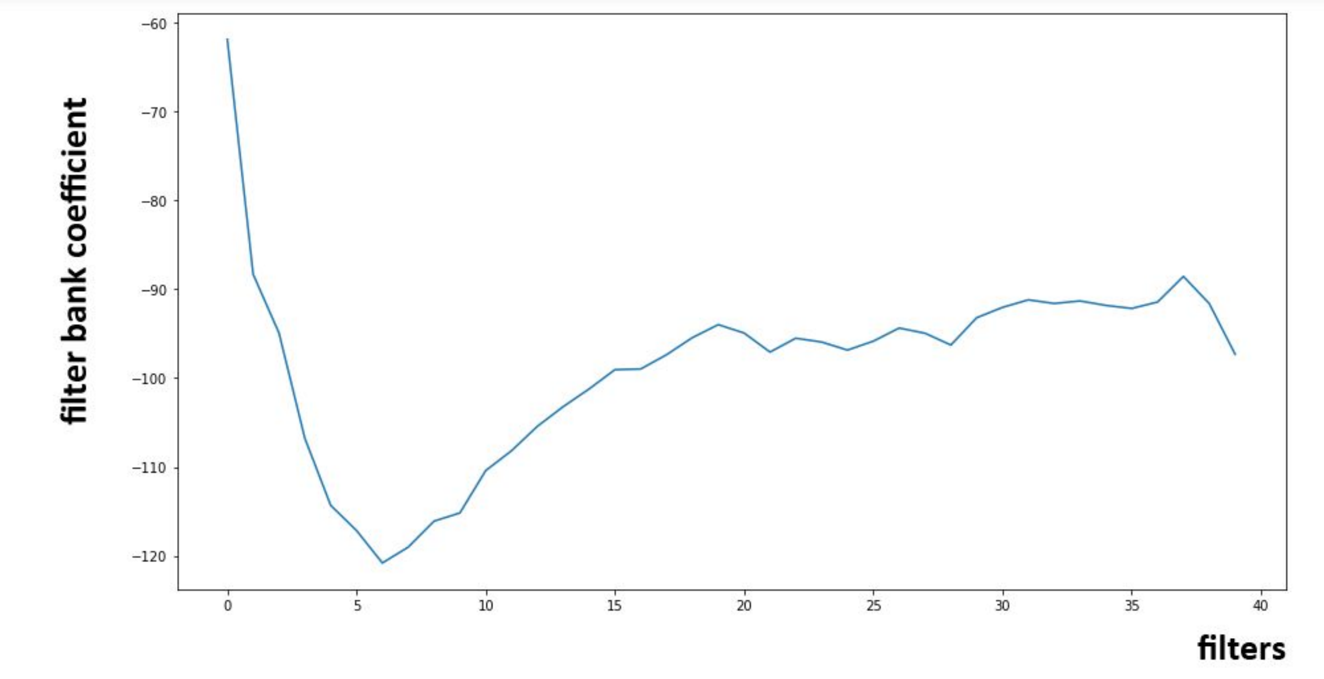


Fig: Applying FFT on the window

Step 8: Discrete cosine transform(DCT)

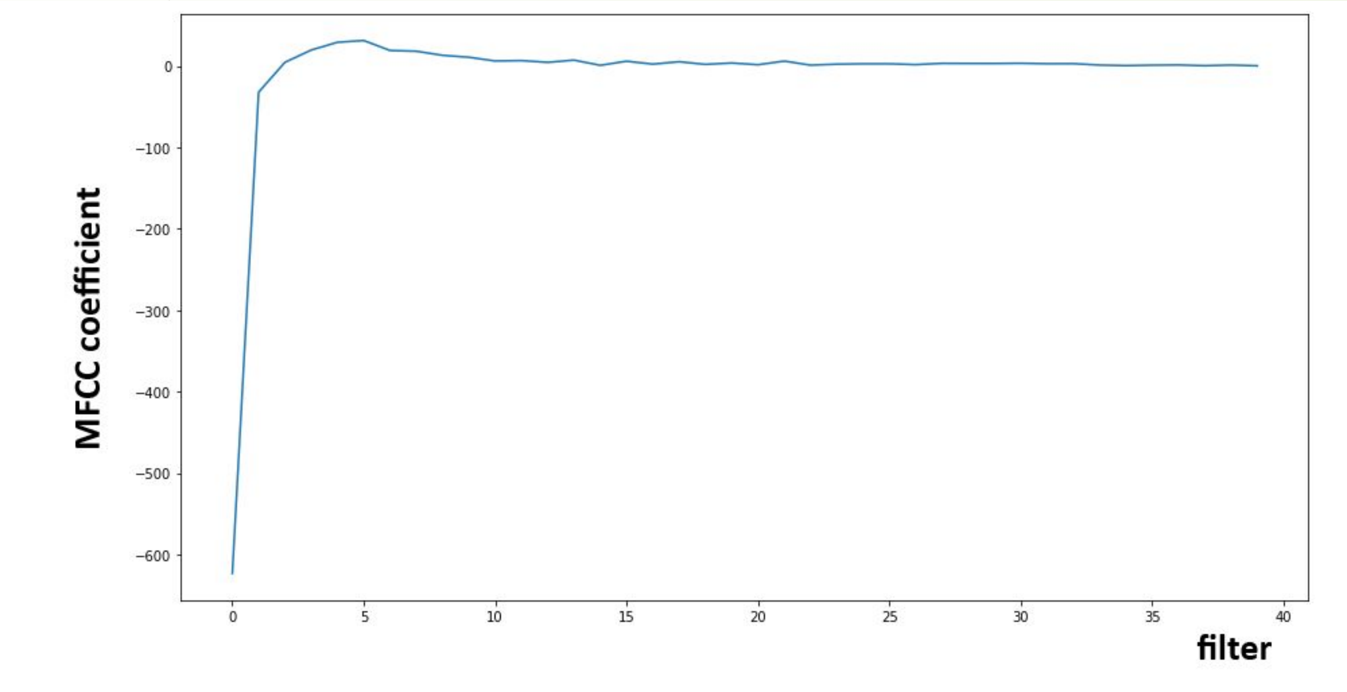
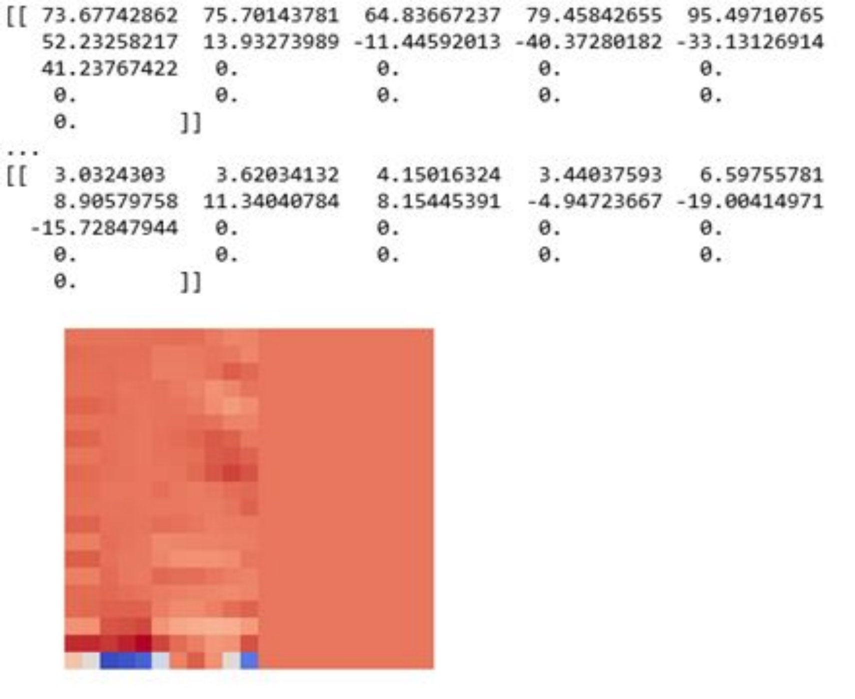


Fig: Applying FFT on the window

MFCC



```
[[ 73.67742862  75.70143781  64.83667237  79.45842655  95.49710765
   52.23258217  13.93273989 -11.44592013 -40.37280182 -33.13126914
   41.23767422   0.           0.           0.           0.
    0.           0.           0.           0.           0.
    0.        ]]
...
[[  3.0324303    3.62034132   4.15016324   3.44037593   6.59755781
    8.90579758  11.34040784   8.15445391  -4.94723667 -19.00414971
  -15.72847944   0.           0.           0.           0.
    0.           0.           0.           0.           0.
    0.        ]]
```

Fig : MFCC encoding showing the vectors and image like data for audio saying “ क “

CNN: Convolution Neural Network

- Used for Analyzing images, classification problems.
- They detect patterns in random images.
- Specifically designed to reduce the image size and extract main features.
- Generates a model.
- Consists:
  - Convolutional layers
  - Pooling Layers
  - Fully connected layers.

CNN

- **Convolutional Layer :**
  - Detects patterns in the image.
  - Each layer is made up number of filters.
  - Filters
    - the matrices that help detect the patterns in an image.
  - Dot product between the filter and same dimensional sections of pixel.
  - This will be the input to next layer and same process will be performed.

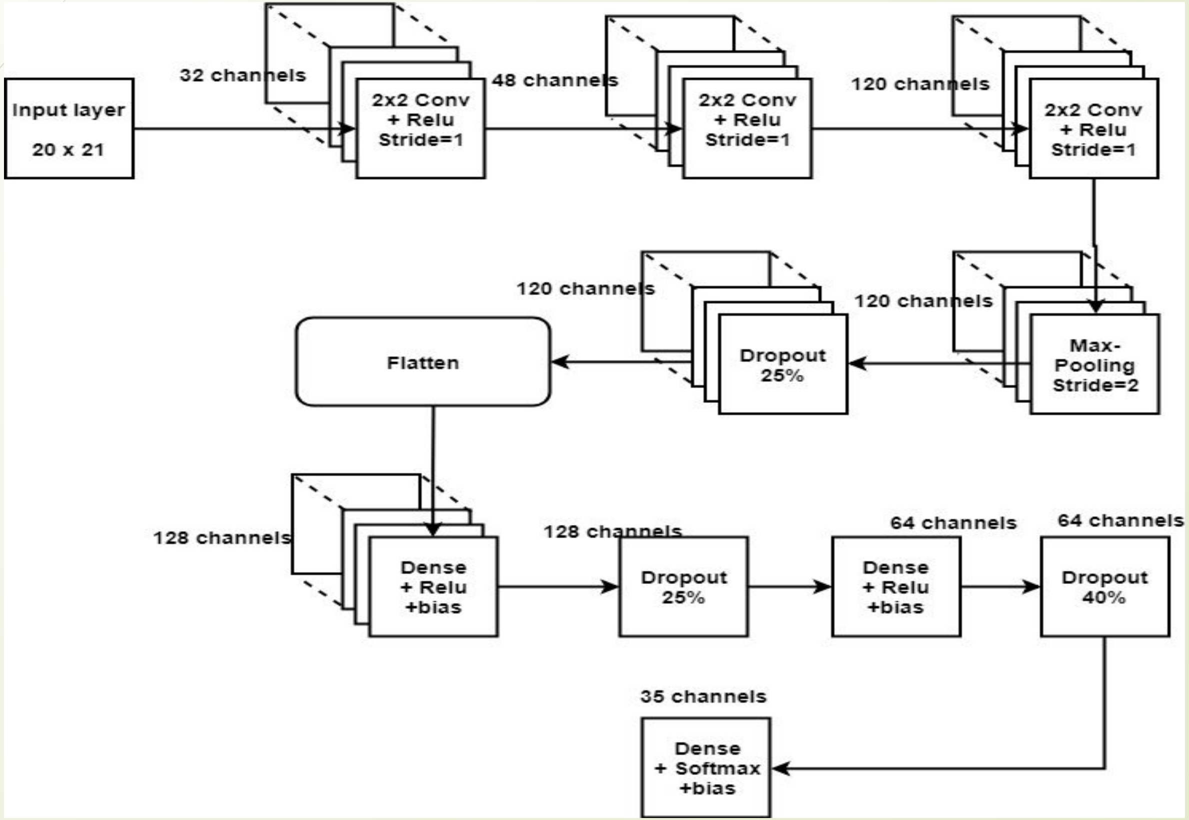
CNN

- **Pooling Layer**
  - A way to take large images and shrink them down while preserving the most important information in them.
  - The output will have the same number of images, but they will each have fewer pixels.
  - Reduces the amount of parameters and computational complexity in the model.
- **Dropout**
  - Prevents overfitting of the data.
  - Ensures Average activation to be constant.

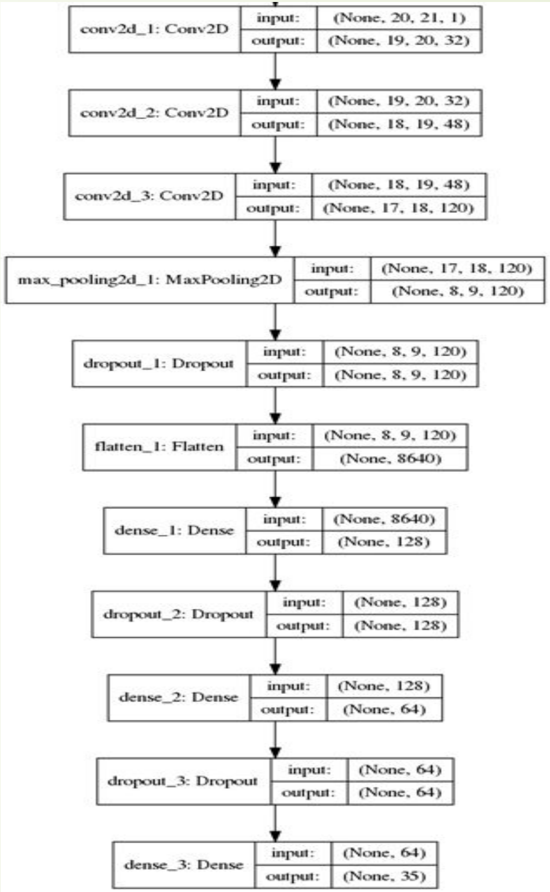
CNN

- **Fully connected layer**
  - Takes output from both convolutional and pooling layers.
  - Uses logic to figure out the image.
- **Activation Functions:**
  - Basically decides whether a neuron should be activated or not.
  - They introduce non-linear properties to our Network.
  - Without it, the output would simple be a linear function.
  - Activation functions used:
    - ReLU
    - Softmax

PROPOSED ARCHITECTURE



CNN MODEL



conv2d_1: Conv2D | input: (None, 20, 21, 1) | output: (None, 19, 20, 32)

conv2d_2: Conv2D | input: (None, 19, 20, 32) | output: (None, 18, 19, 48)

conv2d_3: Conv2D | input: (None, 18, 19, 48) | output: (None, 17, 18, 120)

max_pooling2d_1: MaxPooling2D | input: (None, 17, 18, 120) | output: (None, 8, 9, 120)

dropout_1: Dropout | input: (None, 8, 9, 120) | output: (None, 8, 9, 120)

flatten_1: Flatten | input: (None, 8, 9, 120) | output: (None, 8640)

dense_1: Dense | input: (None, 8640) | output: (None, 128)

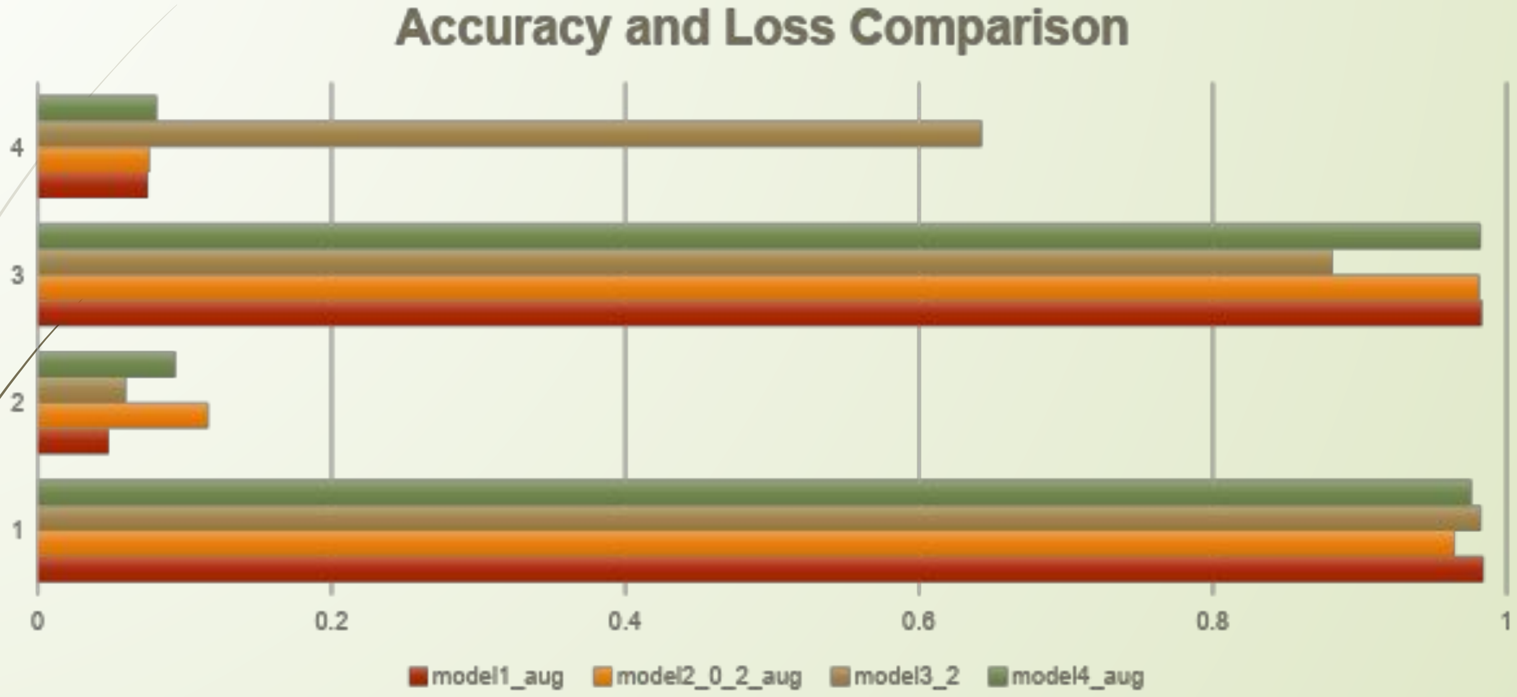
dropout_2: Dropout | input: (None, 128) | output: (None, 128)

dense_2: Dense | input: (None, 128) | output: (None, 64)

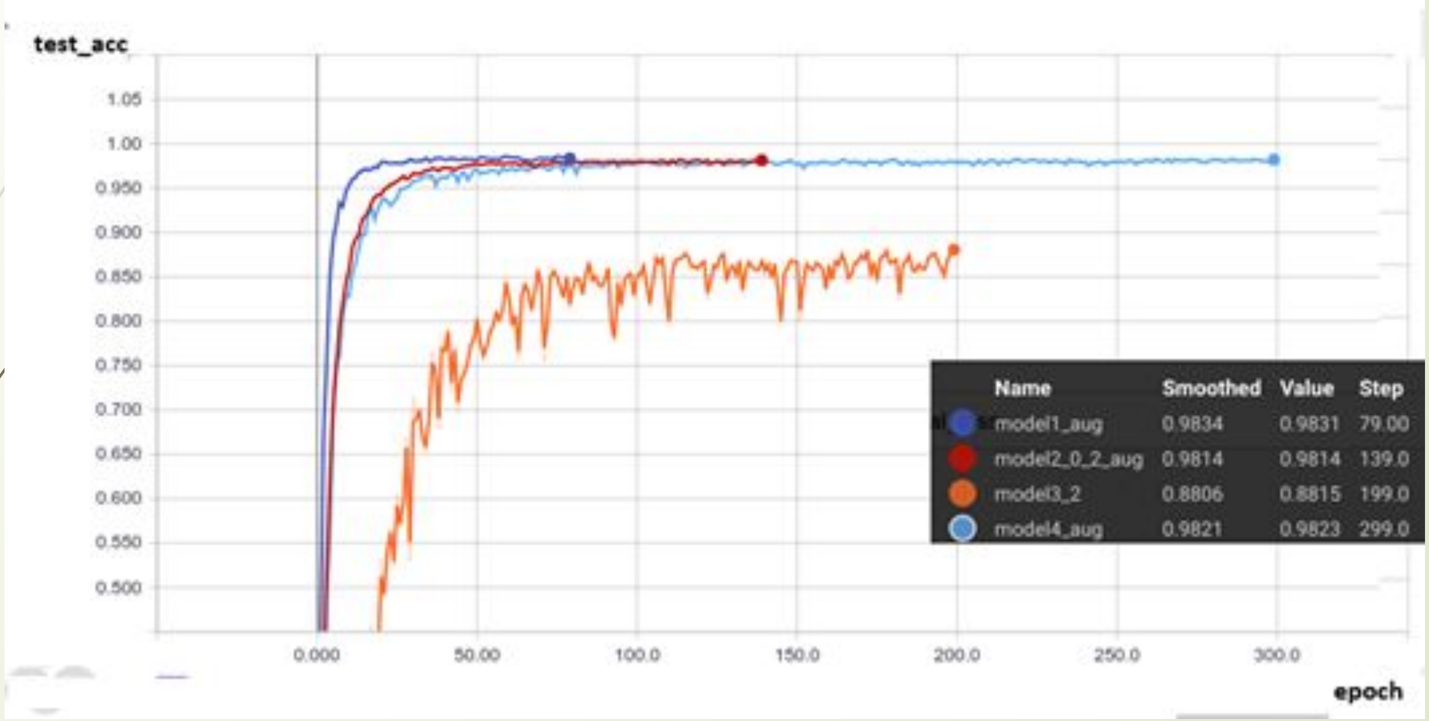
dropout_3: Dropout | input: (None, 64) | output: (None, 64)

dense_3: Dense | input: (None, 64) | output: (None, 35)

Some of the best models:

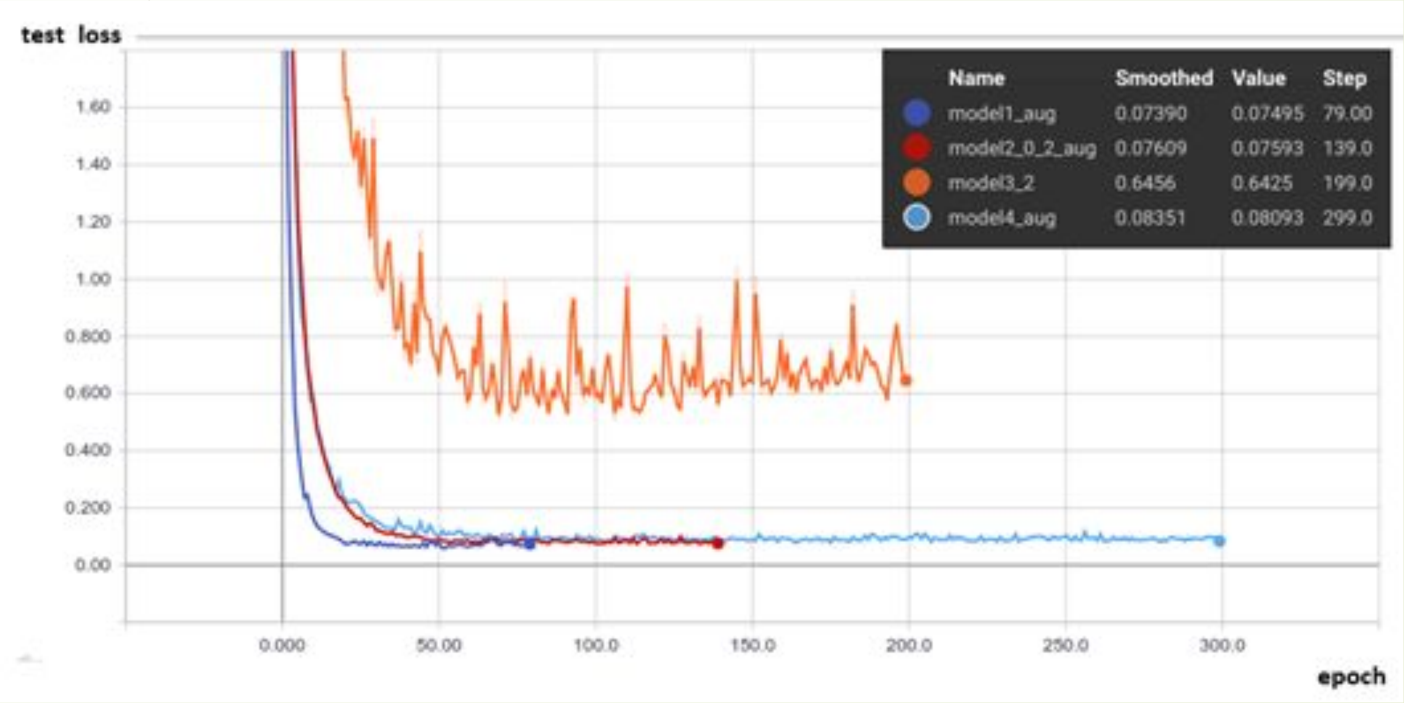


Test Accuracy Comparison



	Name	Smoothed	Value	Step
●	model1_aug	0.9834	0.9831	79.00
●	model2_0_2_aug	0.9814	0.9814	139.0
●	model3_2	0.8806	0.8815	199.0
●	model4_aug	0.9821	0.9823	299.0

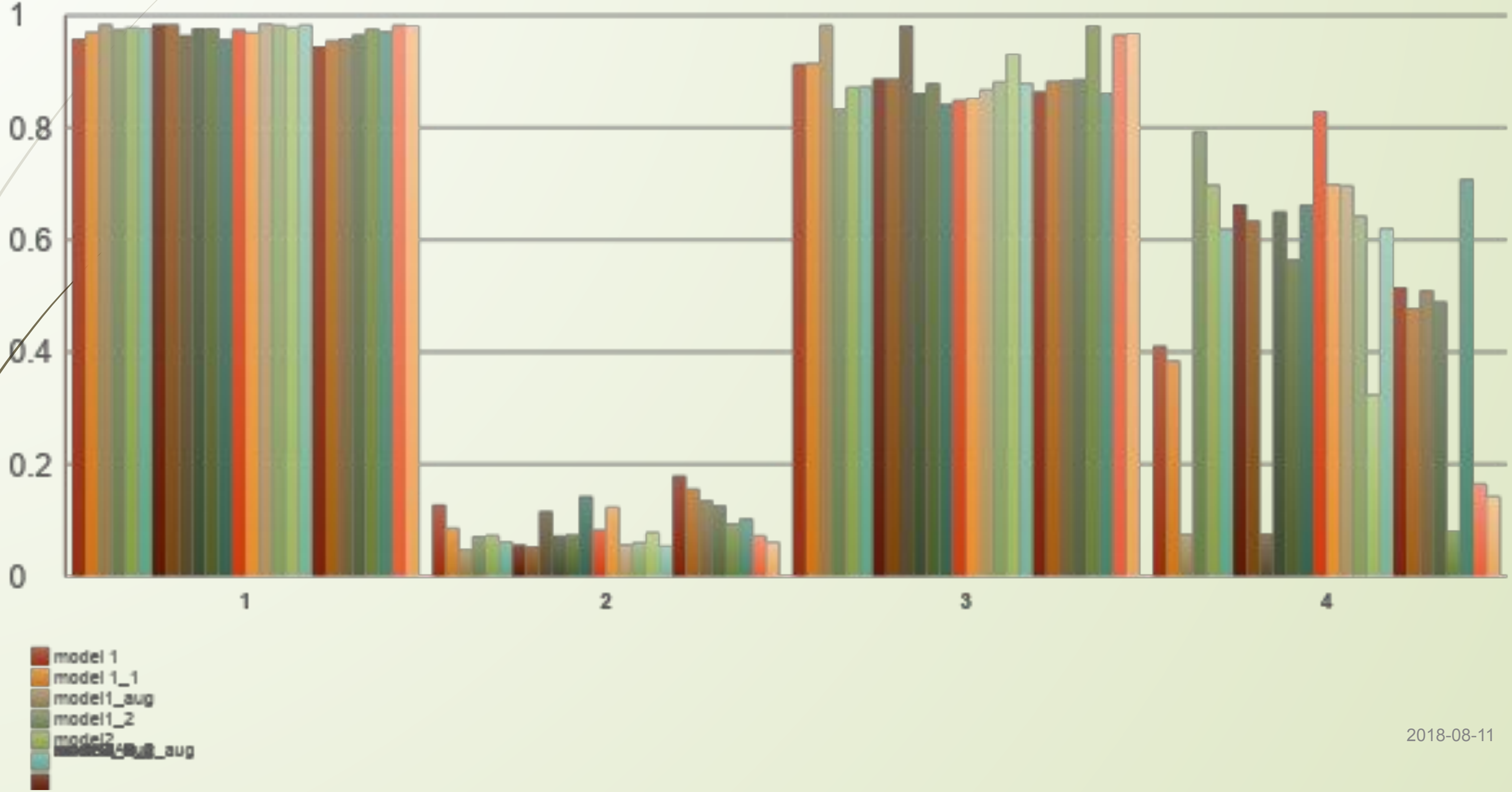
Test Loss Comparison



	Name	Smoothed	Value	Step
●	model1_aug	0.07390	0.07495	79.00
●	model2_0_2_aug	0.07609	0.07593	139.0
●	model3_2	0.6456	0.6425	199.0
●	model4_aug	0.08351	0.08093	299.0

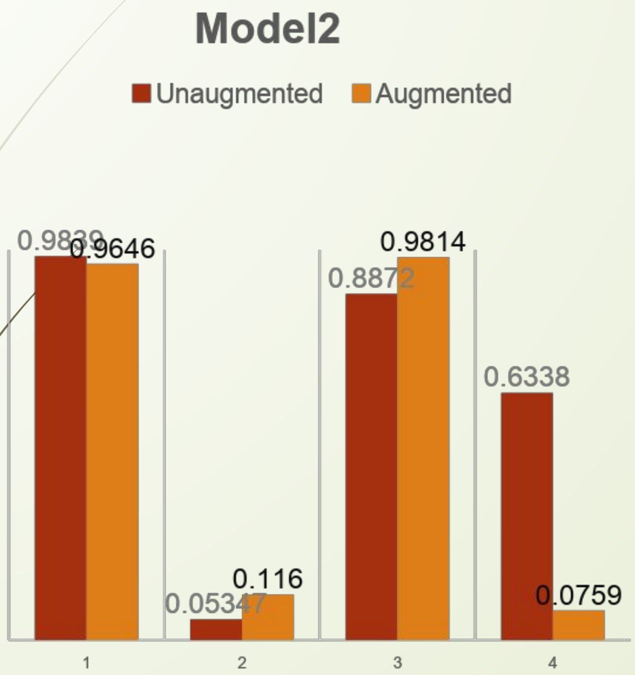
RESULTS

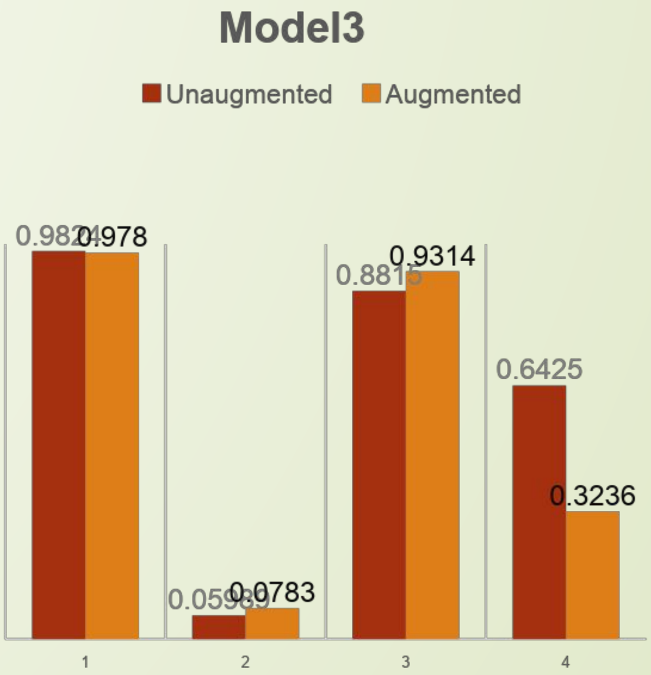
Different Models with altered parameters:



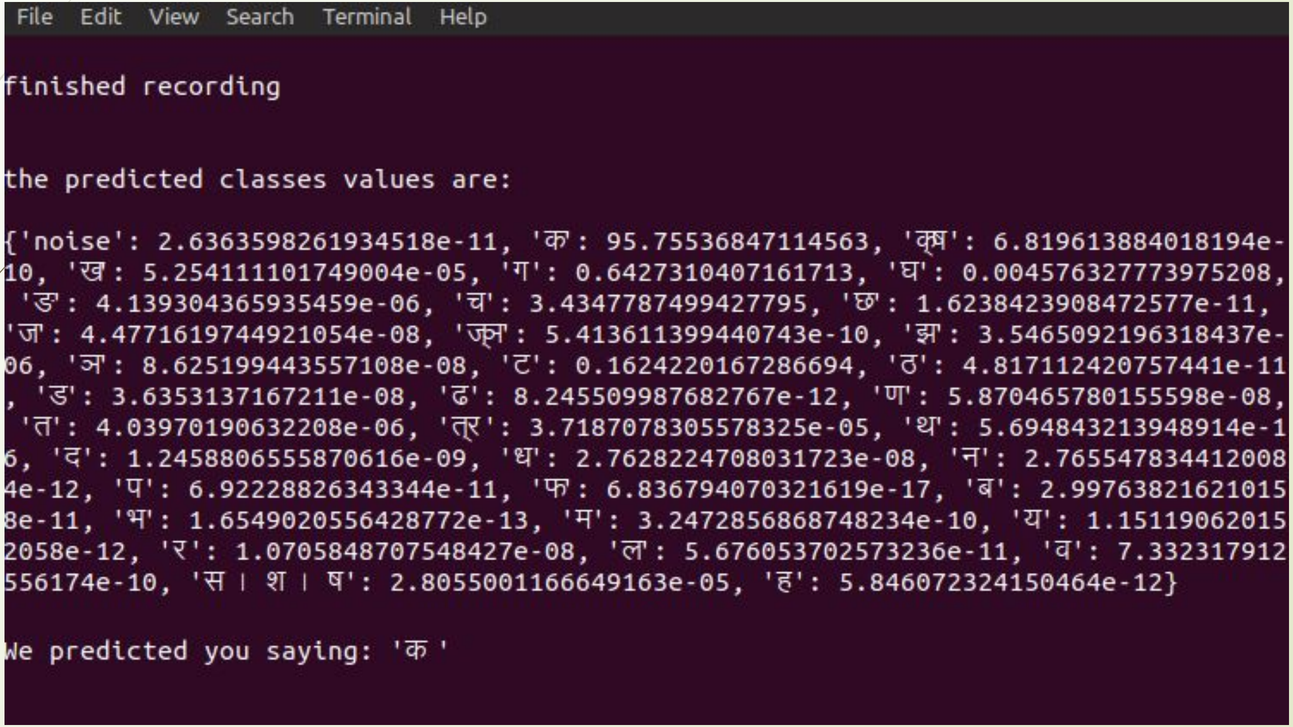
Accuracy and Loss Comparison of Different tested models

Effect of Data Augmentation





Screenshot of Correct prediction of “ka”



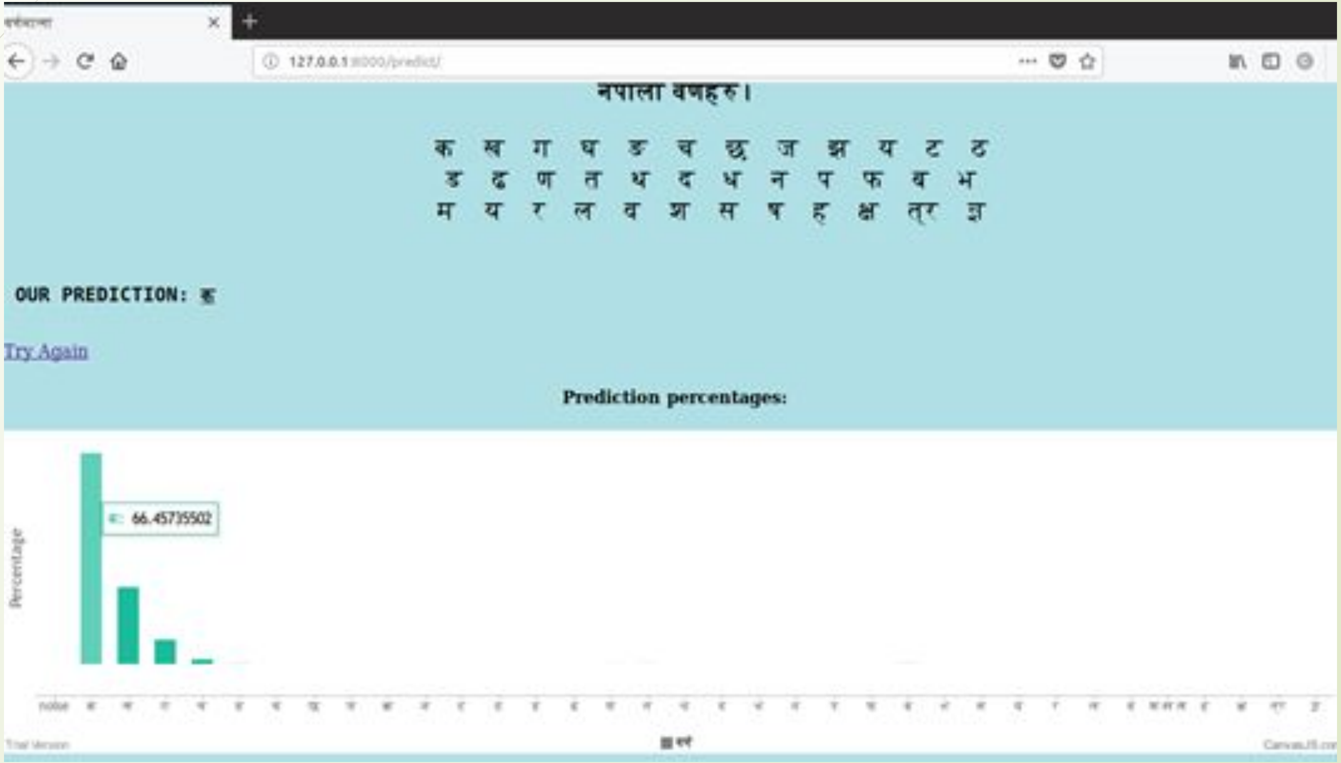
```
finished recording


the predicted classes values are:

{'noise': 2.6363598261934518e-11, 'क: 95.75536847114563, 'क्ष': 6.819613884018194e-
10, 'ख: 5.254111101749004e-05, 'ग': 0.6427310407161713, 'घ': 0.004576327773975208,
 'ङ: 4.139304365935459e-06, 'च': 3.4347787499427795, 'छ: 1.6238423908472577e-11,
'ज: 4.4771619744921054e-08, 'ज्ञ': 5.413611399440743e-10, 'झ: 3.5465092196318437e-
06, 'ञ: 8.625199443557108e-08, 'ट': 0.1624220167286694, 'ठ': 4.817112420757441e-11
, 'ड': 3.6353137167211e-08, 'ढ': 8.245509987682767e-12, 'ण': 5.870465780155598e-08,
 'त': 4.03970190632208e-06, 'त्र': 3.7187078305578325e-05, 'थ': 5.694843213948914e-1
6, 'द': 1.2458806555870616e-09, 'ध': 2.7628224708031723e-08, 'न': 2.765547834412008
4e-12, 'प': 6.92228826343344e-11, 'फ: 6.836794070321619e-17, 'ब': 2.99763821621015
8e-11, 'भ': 1.6549020556428772e-13, 'म': 3.2472856868748234e-10, 'य': 1.15119062015
2058e-12, 'र': 1.0705848707548427e-08, 'ल: 5.676053702573236e-11, 'व': 7.332317912
556174e-10, 'स । श । ष': 2.8055001166649163e-05, 'ह': 5.846072324150464e-12}

We predicted you saying: 'क '
```

Screenshot of Correct prediction of “ka” in web



नेपाली वर्णहरु।

क ख ग घ ङ च छ ज झ ञ ट ठ
ड ढ ण त थ द ध न प फ ब भ
म य र ल व श स ष ह क्ष त्र ज्ञ

OUR PREDICTION: क

Try Again

Prediction percentages:

CONCLUSION

- CNN implemented for Nepali alphabet recognition.
- The trained model can recognize alphabets in an isolated environment.
- Found the importance of data augmentation.

LIMITATIONS AND FUTURE ENHANCEMENTS

- **Limitations:**
  - Lack of datasets.
  - Some similarly sounding alphabet are found hard to detect.
  - Difficulty in detection of alphabets in noisy environment.
- **Future Enhancements:**
  - Datasets can be expanded by collection of voices with a web portal.
  - Continuous speech detection can be implemented.

REFERENCES

[1] R. B. a. S. B. K. H. Davis, "Automatic Recognition of Spoken Digits," J. Acoust. Soc. Am, vol. 24, no. 6, pp. 627-642, 2004.

[2] J. S. a. K. Nakata, "Recognition of Japanese Vowels—Preliminary to the Recognition of Speech," J. Radio Res. Lab, vol. 37, no. 8, pp. 193-212, 1961.

[3] J. S. a. S. Doshita, The Phonetic Typewriter, Information Processing 1962, Munich: Proc. IFIP Congress, 1962.

[4] B. J. a. L. R. Rabiner, "Automatic Speech Recognition – A Brief History of the Technology," p. 6, 2004.

[5] M. Rouse, "TechTarget," [Online]. Available: http://searchcrm.techtarget.com/definition/virtual-assistant. [Accessed 25 December 2017].

[6] I. &. B. Y. &. M. W. &. L. J.-P. Rebai, "Improving speech recognition using data augmentation and acoustic model fusion," j.procs.2017.08.003., vol. 112, pp. 316-322, 2017.

[7] "practicalcryptography," [Online]. Available: http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/. [Accessed 06 03 2018].

[8] cs231n. [Online]. Available: http://cs231n.github.io/convolutional-networks.

THANK YOU !