Bharat Kalra 101803664
Kushagar Sharma 101803672

Speech Recognition Using Convolutional Neural Network

1. Brief Introduction

Emotion Recognition has always been a fascinating problem for researchers. Emotion recognition is a challenging task because emotions are subjective. There are many sub-problems in emotion detection such as emotion recognition from text , emotion recognition from expressions and emotion recognition from speech. This report is about emotion recognition from speech or speech emotion recognition (SER). We define SER systems as a collection of methodologies that process and classify speech signals to detect emotions embedded in them.

2. Related Work

The motivation for this report is Automatic Speech Emotion Recognition Using Machine Learning written by Leila Kerkeni, Youssef serreston, Mohamed Mbarki , Kosai Raoof,Mohamed Ali Mahjoub and Catherine Clender. They computed MFCC and used various state of the art machine learning and deep learning models like SVM, MLR, RNN for classification. In our work we are using Convolutional Neural Netwok(CNN) on MFCC as discussed in further sections.

3. Method and Models

3.1 MFCC

Mel-Frequency Cepstrum (**MFC**) is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency.

Mel is derived as following:

1. Take the Fourier Transform of (a windowed excerpt of) a signal.
2. Map the powers of the spectrum obtained above onto the mel scale, using triangular overlapping windows.
3. Take the logs of the powers at each of the mel frequencies.
4. Take the discrete cosine transform of the list of mel log powers, as if it were a signal.
5. The MFCCs are the amplitudes of the resulting spectrum.

A popular formula for calculating mel value from frequency is:

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right)$$

The purpose for taking log of mel values is to mimic human hearing as human are more sensitive to changes to sound in lower frequencies as compared to Higher frequencies.

Also while extracting windowed excerpt of signal , the signal cannot be just chopped off as it will induce noise at higher frequencies so we want the amplitude of cut signal to drop off at ends that's why we use Hamming window:

$$\textit{Hamming}\ (\alpha = 0.46164)\ \text{or}\ \textit{Hanning}\ (\alpha = 0.5)\ \text{window}$$

$$w[n] = (1-\alpha) - \alpha\cos\left(\frac{2\pi n}{L-1}\right) \qquad L:\ \text{window width}$$

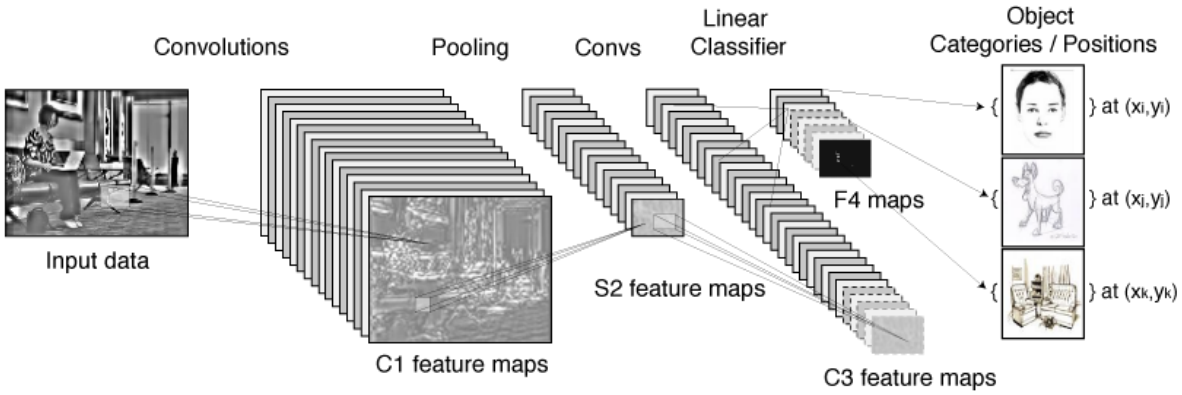
And the formula for chopping is:

$$x[n] = w[n]\,s[n]$$

sliced frame — original audio clip

By doing above process we get a 2d matrix of MFCC values which we store as image. These images are the input of our conv-net.

3.2 Convolution Neural Network



A convolutional neural network consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of a series of convolutional layers that *convolve* with a multiplication or other dot product. The activation function is commonly a RELU layer, and is subsequently followed by additional convolutions such as pooling layers, fully connected layers and normalization layers, referred to as hidden layers because their inputs and outputs are masked by the activation function and final convolution.

Convolution

When programming a CNN, the input is a tensor with shape (number of images) x (image height) x (image width) x . Then after passing through a convolutional layer, the image becomes abstracted to a feature map, with shape (number of images) x (feature map height) x (feature map width) x (feature map channels). A convolutional layer within a neural network should have the following attributes:

- Convolutional kernels defined by a width and height (hyper-parameters).
- The number of input channels and output channels (hyper-parameter).

- The depth of the Convolution filter (the input channels) must be equal to the number channels (depth) of the input feature map.

Convolutional layers convolve the input and pass its result to the next layer. This is similar to the response of a neuron in the visual cortex to a specific stimulus. Each convolutional neuron processes data only for its receptive field. Although fully connected feed forward network can be used to learn features as well as classify data, it is not practical to apply this architecture to images. A very high number of neurons would be necessary, even in a shallow (opposite of deep) architecture, due to the very large input sizes associated with images, where each pixel is a relevant variable. For instance, a fully connected layer for a (small) image of size 100 x 100 has 10,000 weights for *each* neuron in the second layer. The convolution operation brings a solution to this problem as it reduces the number of free parameters, allowing the network to be deeper with fewer parameters. For instance, regardless of image size, tiling regions of size 5 x 5, each with the same shared weights, requires only 25 learnable parameters. By using regularized weights over fewer parameters, the vanishing gradient and exploding gradient problems seen during backpropogation in traditional neural networks are avoided.

Pooling

Convolutional networks may include local or global pooling layers to streamline the underlying computation. Pooling layers reduce the dimensions of the data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Local pooling combines small clusters, typically 2 x 2. Global pooling acts on all the neurons of the convolutional layer. In addition, pooling may compute a max or an average. *Max pooling* uses the maximum value from each of a cluster of neurons at the prior layer. *Average pooling* uses the average value from each of a cluster of neurons at the prior layer.

Fully connected

Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional multi layer perception neural network (MLP). The flattened matrix goes through a fully connected layer to classify the images.

Receptive field[

In neural networks, each neuron receives input from some number of locations in the previous layer. In a fully connected layer, each neuron receives input from *every* element of the previous layer. In a convolutional layer, neurons receive input from only a restricted subarea of the previous layer. Typically the subarea is of a square shape (e.g., size 5 by 5). The input area of a neuron is called its *receptive field*. So, in a fully connected layer, the receptive field is the entire previous layer. In a convolutional layer, the receptive area is smaller than the entire previous layer. The subarea of the original input image in the receptive field is increasingly growing as getting deeper in the network architecture. This is due to applying over and over again a convolution which takes into account the value of a specific pixel, but also some surrounding pixels.

Weights

Each neuron in a neural network computes an output value by applying a specific function to the input values coming from the receptive field in the previous layer. The function that is

applied to the input values is determined by a vector of weights and a bias (typically real numbers). Learning, in a neural network, progresses by making iterative adjustments to these biases and weights.

The vector of weights and the bias are called *filters* and represent particular features of the input (e.g., a particular shape). A distinguishing feature of CNNs is that many neurons can share the same filter. This reduces memory footprint because a single bias and a single vector of weights are used across all receptive fields sharing that filter, as opposed to each receptive field having its own bias and vector weighting.

3.3. Architecture

First of all the audio files are converted to 2d MFCC matrix using the procedure mentioned in section 3.1.

The convnet expects an input of a 50 x 50 x 1 MFCC image of given audio. Then convolutions with 3 x 3 filter size, 32 filters and activation function RELU are performed followed by Maximum Pooling operation with a filter size of 2 x 2 and stride size of 2 followed by another convolution layer with 32 filters and filter size of 3 x 3. The output of convolution layer is then again passed through a pooling layer with a filter size of 2 x 2 and stride size of 2. The resulted output is then passed into an ANN with 512 input units and activation function RELU which is then connected to output layer with 1 unit and softmax activation function.

The optimizer used for training this model is Adam optimizer and loss function used is categorical cross entropy.

4. Result and Discussion

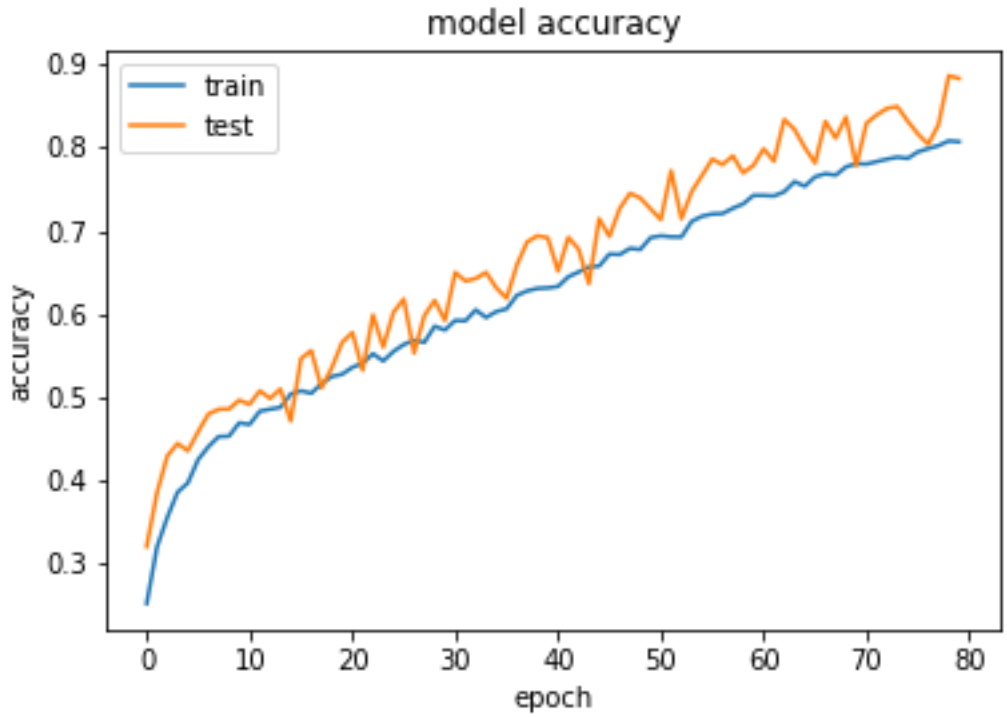
4.1 Dataset

CREMA-D is a data set of 7,442 original clips from 91 actors. These clips were from 48 male and 43 female actors between the ages of 20 and 74 coming from a variety of races and ethnicities (African America, Asian, Caucasian, Hispanic, and Unspecified). Actors spoke from a selection of 12 sentences. The sentences were presented using one of six different emotions (Anger, Disgust, Fear, Happy, Neutral, and Sad).

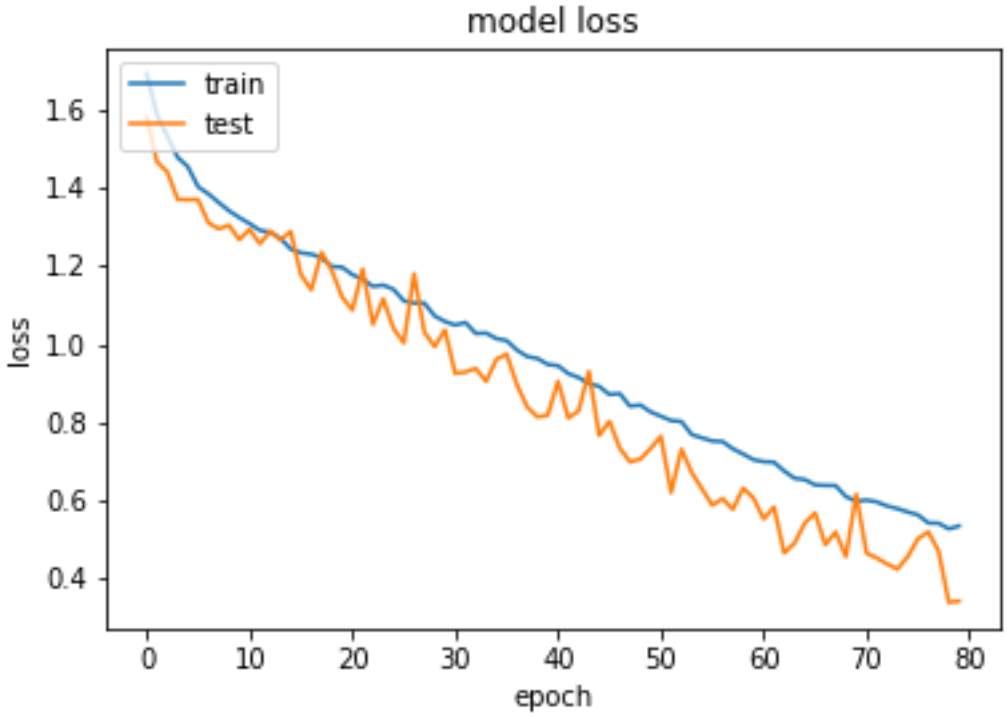
The given dataset is first converted into MFCC and then passed into model. The model is trained for 70 epochs with train test split of 0.2 . The epoch vs accuracy plot of given model is:

Bharat Kalra 101803664
Kushagar Sharma 101803672

The accuracy vs epochs curve is:



The loss vs epochs curve is:



5. Conclusion and Future work

We implemented a pretty robust SER using CNN but there's a lot of room for improvement. Using this system we yet cannot distinguish between male and female voice and the level of the emotion identified.

6. References

[1] Jianxin Wu ,"Introduction to Convolutional Neural Networks" in LAMDA Group National Key Lab for Novel Software Technology Nanjing University, China, May 2017

[2] Leila Kerkeni et `al, "Automatic Speech Emotion Recognition Using Machine Learning"

[3] Ali H, Hariharan M, Yaacob S, AdomAH. Facial emotion recognition usingempirical mode decomposition. ExpertSystems with Applications. 2015;42(3):1261-1277

[4] Surabhi V, Saurabh M. Speech emotion recognition: A review.International Research Journal of Engineering and Technology (IRJET). 2016;03:313-316

[5] Hamdy K. Elminir, Mohamed Abu ElSoud, L. M. Abou El-Maged ―Evaluation of Different Feature Extraction Techniques for Continuous Speech Recognition‖, International Journal of Science and Technology, Volume 2 No.10, October 2012.

Contribution:

Report has equal contribution of both of us

Bharat Kalra: Architecture and Implementation

Kushagar Sharma: Pre-processing, Dataset Modification