Methodology Report

By: Yash Kasundra (a1838670)

University of Adelaide

Applied Machine Learning

ABSTRACT

This report will mainly introduce the methods and metrics used to develop and evaluate this project. The problem statement for the project is to develop a machine learning model that can detect emotions from voice or audio signals. This report will have 2 parts: (i) Machine Learning Methods and (ii) Evaluation Metric.

KEYWORDS

Machine Learning, Neural networks, Speech Emotion Recognition (SER), Convolutional Neural Networks (CNN), Recurrent Neural Network(RNN), Maxpooling, Data Augmentation

1. INTRODUCTION

For this project, my main goal is to create a emotion detecting model that can perceive emotions from any audio signals be it recorded audio or movie dialogue or a real time voice. This is a classification problem where an input sample (Voice signal) needs to be classified into a 8 predefined emotions. And to achieve this I am going to focus on creating my own model using deep neural networks with convolution layers along with maxpooling. The basic process is to train model on my dataset and then fine tune it for better performance. This report will mainly summarize the methodology used in this project and it will also discuss its evaluation metric.

2. MACHINE LEARNING METHODS

In last decade, the deep learning models are gradually replacing traditional machine learning methods and has become the mainstream algorithm in the majority of ML fields. Therefore, several methods that have been implemented and studied.

2.1) RELATED WORK

Since there has been an increase in number of data points or features in audio signal processing domains, many traditional approaches for classification and analysis of audio signals were developed. The main challenge in this project is Feature selection and classification that can accurately identify the emotional state of the speaker [2][3].

Zhang et al. described a method based on the Alex-Net model for emotion recognition in their work [6]. Liu and co. develop a strategy for spontaneously recalling emotions and used it on the RECOLA natural emotion dataset [5]. This research, described an end-to-end LSTM-DNN based model for statistical emotion evaluation that incorporates fully connected layers and the CNN-LSTM technique to extract significant features from raw data [4].

After features extraction from the voice signals, Few of the popular model architecture that has been used over time are:

- **RNN/LSTMs**
- **Attention-based models**
- **Listen-Attend-Spell (LAS)**

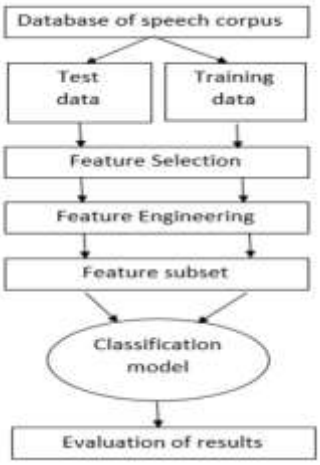

Figure 2: Traditional Machine Learning Approach

2.2) METHODOLOGY

Deep learning methods consists of non-linear components that can perform parallel computations. But to overcome the limitations of different techniques these methods need to be structured with deeper layer of architecture.

Few Fundamental deep learning techniques that are used for SER are Recursive Neural Networks (RNN), Auto Encoder (AE), Deep Belief Network (DBN) , Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN)
can significantly improves the overall performance of the designed system.

For my project, I used these 4 datasets i.e., Surrey Audio-Visual Expressed Emotion (Savee), Crowd-sourced Emotional Mutimodal Actors Dataset (Crema-D), Ryerson Audio-Visual Database of Emotional Speech and Song (Ravdess), Toronto

emotional speech set (Tess),. After merging all these different datasets into single dataframe and classifying all these data into 8 different emotions which are Surprise, Angry, Calm, Disgust, Sad, Fear, Neutral, Happy

I did some preprocessing like data augmenting using noise injection, stretching, shifting and pitch modulating for up sampling the data. I extracted these 5 important features i.e. MelSpectogram , Zero Crossing Rate, MFCC, Chroma_stft , RMS(root mean square) value to train our model.

Now, I had some raw data which needed to be normalized and thus I used oneHotEncoder. Basically, converting the categorical data into numerical data to make it easier for our machine to learn is the task of one-hot encoding. Simply its meaning that if a feature is represented by that column, it receives a 1. Otherwise, it receives a 0 [7].

After normalizing my data with the help of one_hot_encoding, it was time to create train and test dataframes. So I split my current data into 75-25% ratio where 75% was my training set and 25% were my testing data. Then I applied StandardScaler method to remove mean and scale each feature to unit variance. And the main reason for using this method is, since all Features are different their worth to the model is also different. Some Features might contribute more towards the success of model, while some might contribute less. Thus each feature is measured at different scales this might end up creating a bias. To deal with such kind of problem, feature wide standardization (μ=0, σ=1) is used prior to model fitting [8]. After expanding the data to match the input size of our model, it was time to create the deep neural network. To start with I planned on using 1-D Convolutional Neural Networks with relu activation along with maxpooling.

Convolution Layer : Convolution layer handles the main computation of the network and thus it is the core building block of CNN. The basic working of Convolutional layer is that it performs a dot product between 2 matrices, where one matrix is restricted portion of the data and other is the set of learnable parameters sometimes known as kernel [9].

$$W_{out} = \frac{W - F + 2P}{S} + 1$$

Figure 5: Convolution layer formula

Here is an example of how the convolution layer works, it takes a subset of the feature transform it based on the kernel and the convolutional formula and creates a new output as shown below:

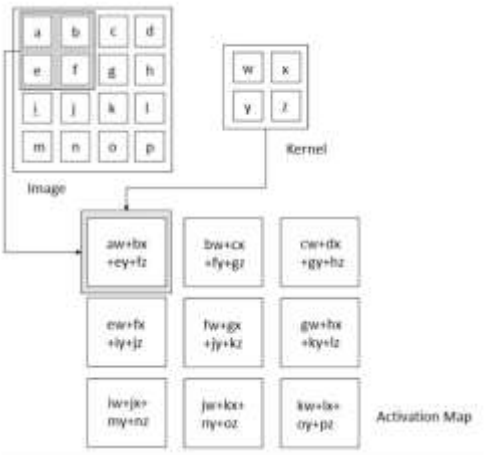

Figure 6: Basic Working of CNN

Pooling Layer: Based on the nearby outputs, pooling layer changes the output of network at certain location based on the pooling layer used. This helps in reducing the required number of computations and weights by reducing the spatial size of the representation. The pooling operation is processed on every piece of the representation separately.

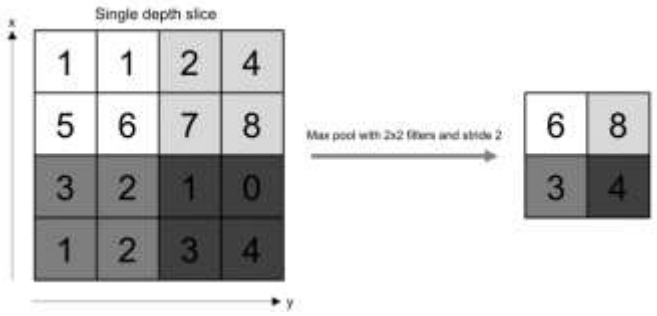

Figure 6: Working of Pooling layer

$$W_{out} = \frac{W - F}{S} + 1$$

Figure 7: Formula for Pooling layer

There are many pooling functions to choose from such as weighted average based on distance from center, mean of the neighborhood, and L2 norm of the rectangular neighborhood. However, the most popular process is taking the maximum output from the neighborhood which is also call max pooling [10].

Below image shows the architecture of my Model, it can change overtime if I find something is not working well but for now, I've come up with this.

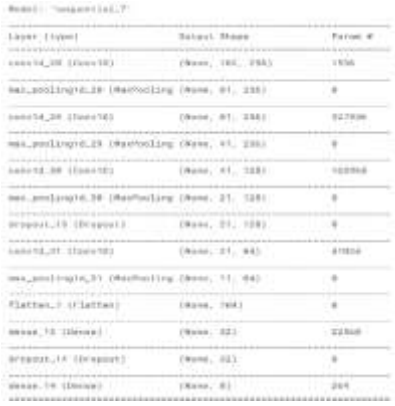

Figure 9: Deep-CNN model

3. Evaluation Metric

If we talk about classification problem, the most common metrics used are:

I. **Accuracy:** It is the most straight forward metric used in Machine Learning. It basically defines how accurate the model is. For example, if the model classifies 90 of the samples from 100 total accurately then it has 0.9 or 90% accuracy.

II. **Precision:** Precision is the ratio of true positives to the total predicted positive observations [11]. Thus, having a high precision relates to low false positive rates. Check figure 10 for more info. There are 2 ways to compute precision for multiclass problem:

- Macro averaged precision: calculate precision for all classes separately and then take their average [11].
- Micro averaged precision: calculate class wise true positive and false positive and then use that to calculate overall precision [11].

Some terminology for evaluation metric are:

True Positive (TP) – These are correctly predicted values which means the actual class was yes and the predicted class was also yes [11].

False Positive (FP) – These are predicted incorrectly i.e. the actual class is no but it is predicted yes [11].

True Negative (TN) – These are correctly predicted negatives that means the actual class was no and the predicted class was also no [11].

False Negative (FN) – The actual class was yes and the predicted class was no [11].

III. **Recall:** It is also known as sensitivity because it is the ration of correctly predicted positive observation to all the observation in actual class [11]. Check figure 10. Similar to precision, we can calculate recall in 2 ways Macro and Micro for multi class problem.

IV. **F1 Score:** F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. F1 score is a metric that combines both precision and recall [11]. It is defined as a simple weighted average (harmonic mean) of precision and recall. Check out figure 10 for the mathematical representation of F1-score. And there are 2 ways to calculate F1 score for multiclass problem i.e.

- **Macro averaged F1 Score**: calculate f1 score of every class and then average them [11].
- **Micro averaged F1 Score**: calculate macro-averaged precision score and macro-averaged recall score and then take there harmonic mean [11].

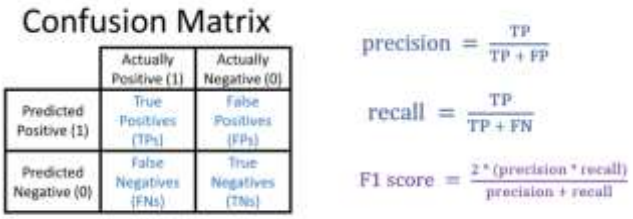

Figure 10: Confusion Matrix and Mathematical representation of Precision, Recall and F1-Score

REFERENCES

[1] Middleton, M., 2022. Deep Learning vs. Machine Learning — What's the Difference? | Flatiron School. Flatiron School. Available at: https://flatironschool.com/blog/deep-learning-vs-machine-learning/

[2] S.R. Ashokkumar, G. MohanBabu
Extreme learning adaptive neuro-fuzzy inference system model for classifying the epilepsy using Q-Tuned wavelet transform
J. Intell. Fuzzy Syst., 39 (1) (2020), pp. 233-248

[3] M. Premkumar, T.V.P. Sundararajan
Defense countermeasures for DoS attacks in WSNs using deep radial basis networks
Wireless Pers. Commun., 120 (4) (2021), pp. 2545-2560

[4] A. Ganapathy
Speech emotion recognition using deep learning techniques
ABC J. Adv. Res., 5 (2) (2016), pp. 113-122

[5] M. Premkumar, T.V.P. Sundararajan, K.V. Kumar
Various defense countermeasures against DoS attacks in wireless sensor networks
Int. J. Sci. Technol. Res., 8 (10) (2019), pp. 2926-2935

[6] M. Premkumar, M. Kathiravan, R. Thirukkumaran
Efficient broadcast authentication using TSG algorithm for WSN
Int. J. Comput. Appl., 58 (4) (2012), pp. 34-39

[7] scikit-learn.2022. sklearn.preprocessing.OneHotEncoder https://scikit-

learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html

[8] Loukas, S., 2022. How Scikit-Learn's StandardScaler works. Medium. https://towardsdatascience.com/how-and-why-to-standardize-your-data-996926c2c832#:~:text=StandardScaler%20removes%20the%20mean%20and,standard%20deviation%20of%20each%20feature

[9] Brownlee, J.,2022. https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/

[10] Le, J., 2022. Convolutional Neural Networks: The Biologically-Inspired Model | Codementor. Codementor.io. https://www.codementor.io/@james_aka_yale/convolutional-neural-networks-the-biologically-inspired-model-iq6s48zms

[11] T, B., 2022. *Comprehensive Guide on Multiclass Classification Metrics* Medium https://towardsdatascience.com/comprehensive-guide-on-multiclass-classification-metrics-af94cfb83fbd