Emotion Recognition in Speech

Nick Mostowich, Ruchir Doshi, Imaad Umar, Shouvik D'Costa
Systems Design Engineering, Department of Engineering
University of Waterloo
Waterloo, Canada

Abstract— Emotional state of a speaker is to be classified using speech (audio) as the only input. A Neural Network system was trained and tested with a feature set consisting of Mel Frequency Cepstrum Coefficients and pitch. The Neural Networks were also tested with a Fuzzified feature set. The system achieved its best accuracy of 51.3% with a simple neural network operating on the raw data without fuzzification.

This project encompasses SYDE522 Course Project Option A with Hybrid Approach

Keywords— *Neural Networks, Fuzzy Logic, Speech, Emotion Recognition, Human Computer Interaction*

I. Introduction

Speech recognition is an important application of machine intelligence techniques that has grown by leaps and bounds in the past decade. A machine able to transcribe spoken word into text used to be in the realm of science fiction, but such capabilities now exist in the smartphones that have become so ubiquitous across the globe. As human-computer interactions increase in complexity and scope, simply being able to understand *what* is said in a given speech sample is inadequate.

When humans communicate with one another, body language and inflection carry a great deal of information about the context of what is being said. The emotional content of a speech sample provides the listener with valuable information about how and why something is being said, in addition to the raw content of the message. Humans use this information to construct a full mental model of the communiqué in order to determine the appropriate response. However, machines do not yet have this capability. This project attempted to build a system capable of analyzing the emotional content of a piece of speech independent of the actual words being said.

This constraint of determining emotion independent of the actual content of a speech sample complicates the problem greatly. Humans are able to cross reference the actual message of the words with the perceived emotion in order to create an accurate and deep understanding of the communication, but determining emotion from speech without any context is quite difficult. For example, if someone says "I am extremely angry" it is quite likely that they are angry, regardless of the emotion present in their voice alone. Still, humans can usually glean emotion from each other across language barriers by analyzing the inflection in the voice, facial expression, and the body language. This proposed solution will not have the advantage of facial expression or body language, but seeks to explore the feasibility of determining emotion from raw speech audio alone, without any other context.

This project evaluates the performance of a neural network when analyzing processed speech signals for their emotional context across six possible emotions: anger, fear, sadness, disgust, surprise, and happiness. Various shapes and sizes of neural network are investigated, and a fuzzification step is implemented in order to evaluate if decreasing the granularity of the input data improves or worsens performance. A neural net was chosen over other types of classifications due to its ability to deal with complex interspersed data in a high number of dimensions, its ability to deal with multiclass classification problems, and the manner in which it models a human brain.

II. State of the Art

Many research groups have attempted to classify emotion of a user's dialogue using speech, facial analysis, and pose analysis. Significant study was made on three papers that cover techniques that use neural networks[1], Hidden Markov Models [2], Bayesian Networks [3], and distance classifiers (Support Vector Machines)[1][3] on both speech and facial analysis.

A. Intelligent Facial Action and Emotion Recognition for Humanoid Robots

Li Zhang et al analyzed a user's face for emotional state. The facial images are analyzed and Action Units are extracted as features. These action units focus on changes in specific parts of the face such as "Inner Brow Raiser". These action units are fed into two neural networks, one for the upper half of the face and one for the lower. Both the outputs are then sent to a Support Vector Machine which gives the final predicted emotional state of the user. This system achieved an accuracy of 80%.

B. Interaction Style Detection Based on Fused Cross-Correlation Model In Spoken Conversation

Wen Li-Wei et al used just speech to look at emotional state. The emotional state classifier was part of a sub system which was used to get the Interaction Style of the user. The emotional classifier was based on an HMM which used the temporal phase of the emotional state as one of the features along with pitch, energy, and formants. The accuracy for the

HMM system was 79.82%. However, the system was not a pure emotional classification based on speech. Before the emotion of the speaker was determined the system was fed the answers they gave to a personality survey, and some baseline answers the subjects gave to some basic questions. Thus the system Wen Li-Wei et al designed was able to accurately classify human emotion of subjects after being finely tuned to the nuances of that specific subject's communication style. While this may model the manner in which humans learn the nuances of each other through interpersonal relationships, it makes the approach unsuitable to unsupervised classification of new, context-free speech samples.

C. *A Real Time and Robust Facial Expression Recognition and Imitation approach for Affective Human-Robot Interaction Using Gabor filtering*

Felipe Cid et al used Bayesian Networks and distance classifiers to classify the user's emotional state by applying edge detection on a user's facial image extracting a feature set. The features are extracted as Action Units which correspond to distortions in the face through muscular activity. These are fed into a Dynamic Bayesian Network which outputs one of five states: happy, sad, anger, fear and neutral. The system achieved an accuracy of 90%.

III. Problem Statement

Traditional speech recognition software analyzes the content of speech without understanding the context. The context is usually understood through the emotion conveyed, and as a result traditional speech recognition fails to interpret important information for human communication. This project attempts to glean the emotional characteristic of spoken English, independent of content. The emotion of a speech sample is modeled as a discrete state such that a subject can only contain one emotion at once; for example a user cannot be both happy and disgusted at the same time.

IV. Proposed Solution

Based on prior art many researchers used a Hidden Markov Model (HMM) to tackle this problem. However these researchers used HMM in conjunction with audio and video samples of human emotional speech. This project was scoped to only use audio samples of emotional speech and therefore using HMM would not be ideal as it uses a state to state transition model. With video sampling it is relatively simple to analyze the changing state of a subjects emotion based on facial cues, and thus a state to state transition model makes sense, but for a system that intends to classify one emotion per entire sample an HMM is not applicable. The approach was revised to analyze speech by using a Neural Network.

The high level description of our process is as follows:

1. Save all speech data as wave (.wav) files
2. Convert audio data from time domain to frequency domain and extract pitch
3. Extract the following 11 features (per sample window of 25 ms) from the pitch for each file
    a) Mean
    b) Variance
    c) Min
    d) Max
    e) Median
    f) Mean derivative
    g) Variance derivative
    h) Min derivative
    i) Max derivative
    j) Median derivative
    k) Spurt length
4. Convert frequency data to Mel Frequency Spectrum
5. Obtain 13 Mel Coefficients (per sample window of 25ms)
6. Extract the following 5 features for each of the 13 Coefficients and its time derivative; this yields 130 features.
    a) Mean
    b) Variance
    c) Min
    d) Max
    e) Median
7. Combine all features for a total of 141
8. Optionally fuzzify feature set (see below)
9. Train Neural Network on the 141 features

After deciding on the model of the solution the next biggest milestone involved getting test and training data for the system. Open source emotional databases are difficult to find and the initial approach was to find audio samples from YouTube videos. Some qualitative tests were run on the YouTube audio clips however, it was determined that the YouTube samples were both too noisy for proper training and evaluation, as well as too time consuming to find. YouTube clips generated often contained considerable cross chatter or background noise, and were often a low quality audio signal. The main emotional speech audio clips used in this report were thus based on the eNTERFACE database [4]. This database consists of 44 test subjects that recorded five sentences for each of the six emotions described above in video format. The database is not ideal as it uses subjects from 14 different nationalities all of whom have different accents. Another limitation of the database was the uneven distribution of the genders as 81% of the subjects were male while 19% were female. In addition the database is intended for use by multimodal classification systems that analyze facial information along with or instead of speech; by converting the .avi files to .wav much of the original information contained in each sample is lost.

After obtaining the data the experimental setup involved extracting the pitch and the Mel Frequency Cepstrum Coefficients (MFCC). A collection of MFCCs make up an MFC. For an MFC, the frequency bands are equally spaced on

the mel scale, which is used in speech recognition software as it is a good approximation of human auditory system's response. This project leveraged the use of Voicebox library [5] to extract the pitch and the RASTAMat [6] library to extract the MFCC.

Originally the system transformed the Mel Frequency signal back into a .wav file, from which the same features were extracted as were extracted for the pitch of the non transformed signal. These results were unsatisfactory, however, as the accuracy achieved on the test data hovered between 22% and 40% for all tested neural net sizes. As a result, the following approach was implemented.

From the pitch and derivative of the pitch, the mean, median, min, max, and variance of the signal were extracted. As well the spurt length was analyzed from the pitch, which is a measure of the overall rate a person is speaking at. From the pitch and its derivative a total of 11 features can be extracted and analyzed. Extracting the MFCC and its derivative is a bit more complex as it involves obtaining 13 MFCCs from every 25ms time window in an entire .wav file. Therefore in total there are 130 features extracted from the MFCC and its derivative. In total there are 141 features that can identify and distinguish each of the six emotions.

These features are fed into the Neural Network training system which used 70% of the audio data for training, 15% for validation, and 15% for testing. The data that is randomly selected for training, validation and testing. This means that every time a neural net is trained, the training data selection could be selecting samples that are only male or more of just one emotion. As a result, the confusion value results for test and validation can be significantly skewed. The results are a lot less repeatable due to this process. Since the data actually used for training and testing is different for each network trained, the results are not directly comparable against one another. Still with the exhaustive search process detailed below, and a set of 1263 feature vectors being fed into the system, the negative effects of randomization are minimized.

To test the performance of the system with less granular data, the features were "fuzzified" or bucketed into five separate bins based on their value. The process to do this is as follows. The feature set contained 141 feature columns and n rows corresponding to each sample. The "fuzzification" was implemented column by column. The column was copied to a temporary vector, sorted in ascending order and then separated into five sections. Each section's mean and variance was calculated and then used to produce a normal probability distribution function (PDF). This gave five normal distribution functions for each feature vector which essentially act as the "membership function". Each data point (row) from the feature vector is inputted into the five normal PDFs to obtain the five "memberships". The highest "membership" is chosen to be the bucket and the original data point is then replaced by the "fuzzy value" corresponding to the bucket label which range from 1 - 5. This was repeated for each column in the training data and then fed into the neural network system.

In summary the audio data was analyzed using 11 features collected from the pitch and the derivative of the pitch. From the pitch the mean, median, min, max, variance, and spurt length were calculated. From the derivative of the pitch the mean, median, min, max, and variance were calculated. The second iteration for improving accuracy of the neural network involved the audio files being analyzed using the added 130 features from the MFCC and its derivative. The final iteration for improving accuracy involved using fuzzified input data for the neural network. The accuracy of each approach are shown from confusion matrices in the Results section.

V. Results

The fuzzified training dataset and standard training dataset were tested against a large array of neural net designs. An iterative process that increments the layer sizes (number of hidden units) by ten was used on both single hidden layer networks and two hidden layers networks. For the three hidden layer neural net, the size of the network was increased by 50 with every iteration. The smallest perceptron size was 50, whereas the largest perceptron size used was 250. The resulting number of errors when each network was tested against the test set was saved to a file in the form of confusion values representing the total error rate, from which the best results were analyzed. The full set of confusion matrices against the training data, test data, validation, and all data were then generated for each of these best results to analyze the network for overfit versus true classification performance.

The main advantage of this iterative technique was that it exhaustively tested a large number of network sizes. It enabled analyzation of the system performance in terms of both the efficacy of the fuzzification step and the efficacy of various sized neural nets.

However, the disadvantage of an iterative evaluation process is the sheer run time required. Training and evaluating hundreds of neural networks required many hours of computation. The group only had access to laptops with two or four processing cores and a limited amount of RAM; this disadvantage could be mitigated greatly by parallelizing the evaluation process in the cloud.

Another way to improve the evaluation performance would be to use a smarter evaluation algorithm. A gradient descent search or even simple hill climbing algorithm could greatly decrease the number of networks actually trained. Alternatively an algorithm such as a swarm intelligence or genetic algorithm could help generate optimal network sizing, but would likely be excessive for this kind of problem.

Overall this approach fit well with the project objectives. It allowed testing of the approach across a variety of network sizing, and the efficacy of the fuzzification technique. The first network tested was a simple network operating on

just the feature vector from pitch extraction. As the input layer only contained 11 nodes, the network was quite small.

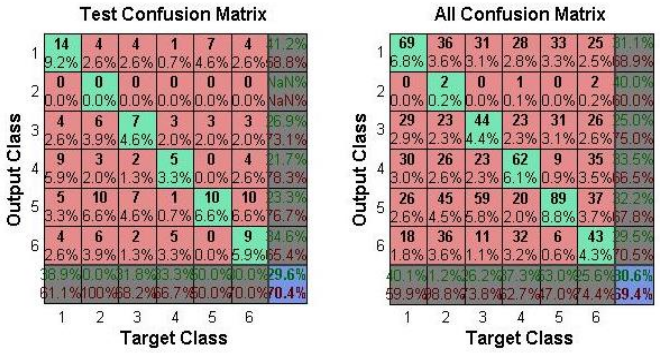

Figure 1: Single Layer, Pitch Only (10 Perceptrons)

The overall accuracy of this approach was quite poor. In the set of test data, only 29.6% of samples were classified accurately. This is still much better than the 16.67% that would be expected from a pure random search, but shows that pitch alone is not enough to detect emotion. The next set of tests were conducted on the raw feature vector using crispy and fuzzy values, across various network sizes. As the feature vector was now a full 141 features, it was expected that the layer sizes producing optimal results would be much larger. This was proven to be the case.

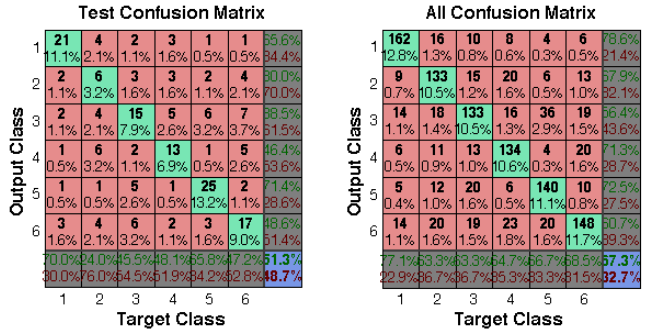

Figure 2: Single Layer (290 Perceptrons)

For the single hidden layer, 290 perceptrons was determined to be the optimal number. This produced much better results than the single layer system without the MFCC vectors. The accuracy on the test set was 51.3%, while across all data the accuracy was 67.3%.

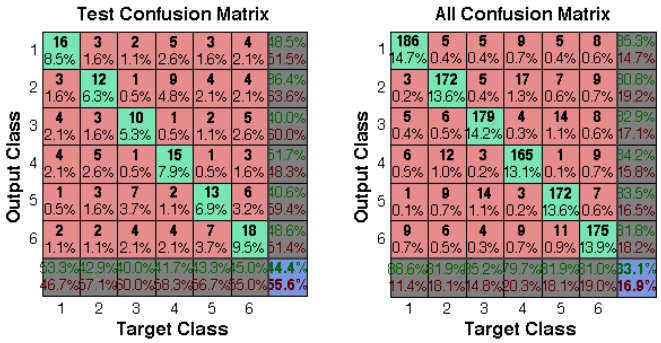

Figure 3: Single Layer with Fuzzification (160 Perceptrons)

The next step was to try a single layer on the fuzzified set of input data. For this test, the optimal layer sizing was determined to be a mere 160 perceptrons. Intuitively this makes sense, as a less complicated set of input data would require fewer neurons to process. The results were quite disappointing, as the fuzzification step actually decreased the accuracy on the test data to 44.4%. Still, it was noticeably faster to both train and classify samples. The amount of arithmetic required on simple integer input is much less than with numbers of very high precision.

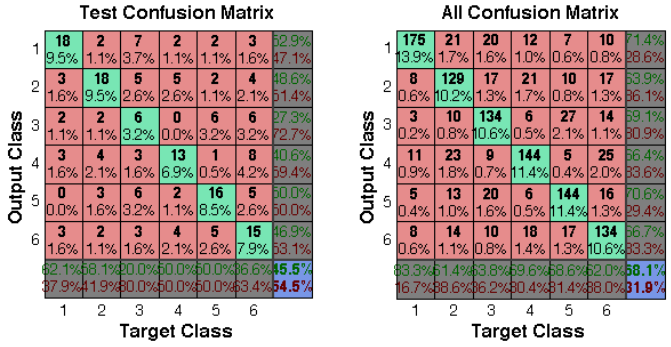

Figure 4: Two Layers (110 perceptrons, 230 perceptrons)

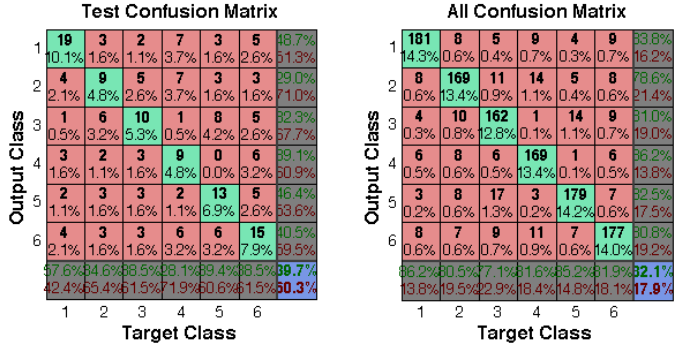

Figure 5: Two Layers with Fuzzification (110 perceptrons, 190 perceptrons)

After the results of the single layer tests, two hidden layers were evaluated. For both the fuzzified and crispy feature

vectors, 110 perceptrons in the first hidden layer was optimal, while 230 and 190 were optimal in the second layer for the crispy and fuzzy sets respectively. The performance of two layers, however, was worse than with a single layer. The standard input data achieved 45.5% accuracy on the test data, while the fuzzified system achieved only 39.7%.

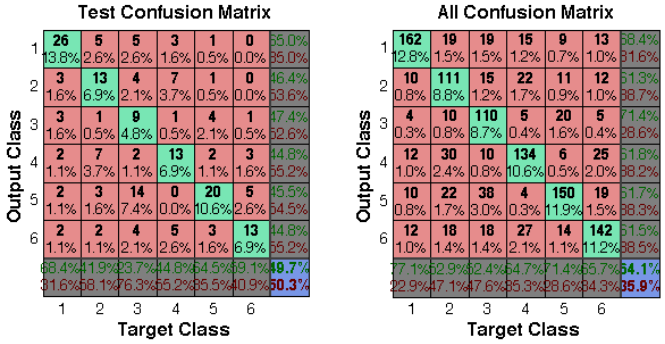

Figure 6: Three Layers (200 perceptrons, 50 perceptrons, 50 perceptrons)

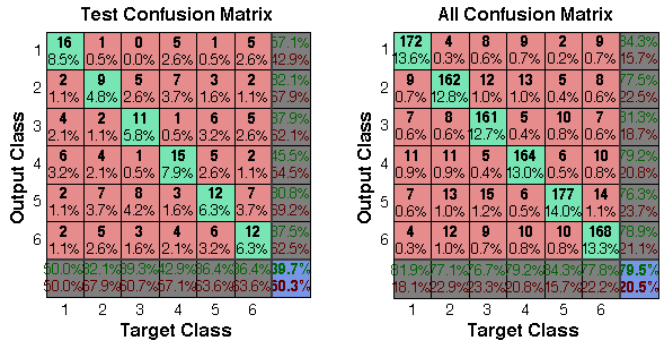

Figure 7: Three Layers with Fuzzification (100 perceptrons, 250 perceptrons, 250 perceptrons)

Testing with three layers changed the results from two layers only slightly. The fuzzy input data achieved 39.7% accuracy, while the crispy values gave a result of 49.7%. The optimal sizing for the crispy values was 200, 50, and 50 perceptrons in the first, second, and third layers respectively while the fuzzy system was optimal at 100, 250, and 250.

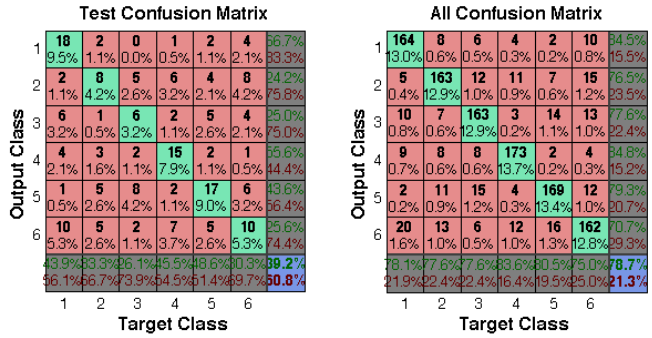

Figure 8: Three Layers with Fuzzification (1000 perceptrons each)

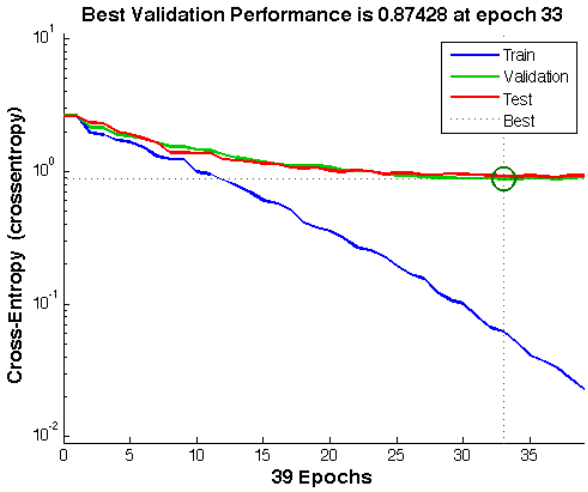

Figure 9: Performance Plot of 1000x1000x1000 Neural Network

The final test conducted, out of pure curiosity, was a neural network with three layers each with 1000 perceptrons operating on the fuzzy data. Training such a system took over ten minutes, but the results were quite interesting. The accuracy on the training data was impressive, at over 99%, but this was clearly the result of overfit. The accuracy on the test data was only 39%. As seen in the performance plot, the system continued to train itself against the training data to the point that it had nearly no errors, but the performance on the actual validation and test sets remained constant. Intuitively this is a sensible outcome; a massive network such as this will learn to recognize specific input vectors, not the actual features contained within those vectors.

VI. Conclusion

There were three main takeaways from this project. First and foremost, emotion detection from speech alone without the content of the speech is an extremely difficult problem. Qualitatively, even the human group members were unable to classify the emotion of speech with a high degree of accuracy; quick tests conducted at the end of the project showed that humans were only able to achieve approximately 50% accuracy, even with the content of the speech being understood. This is very close to the results of the computational system, and thus an accuracy rate hovering around 50% is comparable to human performance and should not be considered a failure. Secondly, the single layer neural nets performed the best. A large hidden layer of 290 perceptrons was able to achieve classification accuracy above 50%, while no other system was able to break this benchmark number. The multi layer networks performed better on the training set, but much worse against the test and validation sets. The effects of overfit are clear. Finally, while the fuzzification step led to a large performance increase in the time required to train and classify with a neural network, it decreased the overall accuracy.

There are thus a few main recommendations for future systems designed in the same vein, summarized in point form below.

- Better training data from more varied subjects and split more equally amongst the genders would enable better system accuracy and reduce the effects of overfitting
- Adding the actual content of the speech via keyword recognition would likely add a feature vector strongly correlated with emotion that would more closely approximate how a human listener determines speech emotion
- Iterating on and improving the fuzzification system may improve accuracy; perhaps five discrete values is not the optimal number to use, and perhaps the Gaussian distributions are not the best way to create the membership functions
- The granularity of the MFCC extraction system could be increased. 13 coefficients may be too few. In addition, only the coefficients and their first derivatives were used in classification. Adding the second or even third derivatives would provide more features that could improve performance

In short, this project investigated the difficulties and nuances of emotion recognition from speech without content, and showed that while accuracy comparable to a human subject is possible, the results are not yet accurate enough to implement in real world systems.

VII. References

[1] Zhang, Li, Alamgir Hossain and Ming Jiang. "Intelligent Facial Action and Emotion Recognition for Humanoid Robots." 2014 International Joint Conference on Neural Networks (IJCNN) . Beijing, China, 2014.

[2] Wei, Wen-Li, et al. "INTERACTION STYLE DETECTION BASED ON FUSED CROSS-CORRELATION MODEL IN SPOKEN CONVERSATION." International Conference on Acoustics, Speech and Signal Processing (ICASSP). Tainan, Taiwan, 2013. 8495-8499.

[3] Cid, Felipe, et al. "A Real Time and Robust Facial Expression Recognition and Imitation approach for Affective Human-Robot Interaction Using Gabor filtering." 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Tokyo, 2013.

[4] Enterface.net, 'eNTERFACE'05: The SIMILAR NoE Summer Workshop on Multimodal Interfaces', 2015. [Online]. Available: http://www.enterface.net/enterface05/main.php?frame=emotion. [Accessed: 05- Apr- 2015].

[5] Ee.ic.ac.uk, 'VOICEBOX', 2015. [Online]. Available: http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html. [Accessed: 05- Apr- 2015].

[6] Labrosa.ee.columbia.edu, 'PLP and RASTA (and MFCC, and inversion) in Matlab using melfcc.m and invmelfcc.m', 2015. [Online]. Available: http://labrosa.ee.columbia.edu/matlab/rastamat/. [Accessed: 05- Apr- 2015].