Multimodal Emotion Recognition

Colin Grubb

March 21, 2013

Abstract

Multimodal fusion is the process whereby two or more forms of input are gathered together in order to produce a higher overall classification accuracy than individual unimodal systems. This is a popular technique in emotion recognition. In this study, we attempted to discover how much we could improve upon individual unimodal systems using decision level fusion. To accomplish this, we acquired two emotion classification systems, one that worked on audio input alone and another that worked on visual input, and combined their output using a set of manual rules and a classifier to achieve higher classification accuracy.

1 Introduction

Machines are becoming more integrated into our society, and robots and computers are starting to become more than just tools that assist us. They can beat us at games and puzzles and allow us to solve massively complex problems using their computation power. Ultimately, if robots and machines are to become integrated into our society to a point where we are interacting with them as humans interact with each other, then computers need to be able to perform some of the same social skills as humans. Humans are capable of recognizing another human's emotions, and they also possess the ability to process information from more than one input (for example, an angry person raising their voice and their expression contorting and their face turning red), and produce a single conclusion or response to those inputs. We could have machines that perform complex tasks such as psychiatry, or robots completely handling customer service rather than eventually transferring to a human if the user becomes too angry. If we want systems that can perform these types of tasks where emotion is important, then machines need to be able to perform the same fusion process that humans can.

Emotion recognition is the act of classifying a person's emotional state based on input from that person. This has been a long interest of artificial intelligence and machine learning research. It has been the focus of various previous studies [9] [8], and research has looked at visual, audio, and gesture information individually. A system that performs emotion recognition on one form of input [ex: audio input only] is considered a unimodal system, and audio information is considered a single "modality". A relatively new sub-field of emotion recognition is the interest in multimodal fusion, and a system that works one two or more modalities [ex: audio and face information, face and gesture information] can be considered a multimodal system. In this study, we seek to ascertain how we can improve emotion recognition by using two existing unimodal systems and combining them using a classifier at the decision level to create a final classification. As a smaller side goal, we aim to combine the software in such a way that the processing of simultaneous input will be fully automated. Section 2 will introduce background information relevant to our study. In Section 3, we will discuss the unimodal software systems that we used in the project. Sections 4 and 5 will cover data gathering and the system layout. Sections 6 through 8 contain our various experiments and their results, and Sections 9 and 10 conclude the paper.

2 Background and Motivation

In this section, we discuss some of the background on previous multimodal fusion as well as what motivated our study. Some more research was necessary in order to learn more about multimodal fusion, and to get an idea of what current systems do when combining unimodal systems for their research purposes. The first resource was a paper called "Analysis of Emotion Recognition using Facial Expressions, Speech and Multimodal Information"[6]. Zhigang Deng and Carlos Busso discussed two main methods of combining unimodal systems: "feature-level fusion, in which a single classifier with features of both modalities is used, and decision level fusion, in which a separate classifier is used for each modality, and the outputs are combined using some criteria". Another study was conducted in 2007 using face and audio information, as well as gesture information [3]. This study also looked at decision and feature level fusion, finding that feature level fusion was more effective than the simple rule set they developed for decision level fusion.

It was decided that we would implement decision level fusion. For this study, we assumed that decision level fusion would be easier to manage using off-the-shelf unimodal components, and since a simple set of rules seems to less effective than feature level fusion, we want to experiment with using a classifier, in place of a set of rules, trained on the outputs and probabilities of the unimodal systems, and use that to produce a final output.

3 Unimodal Systems and Other Software

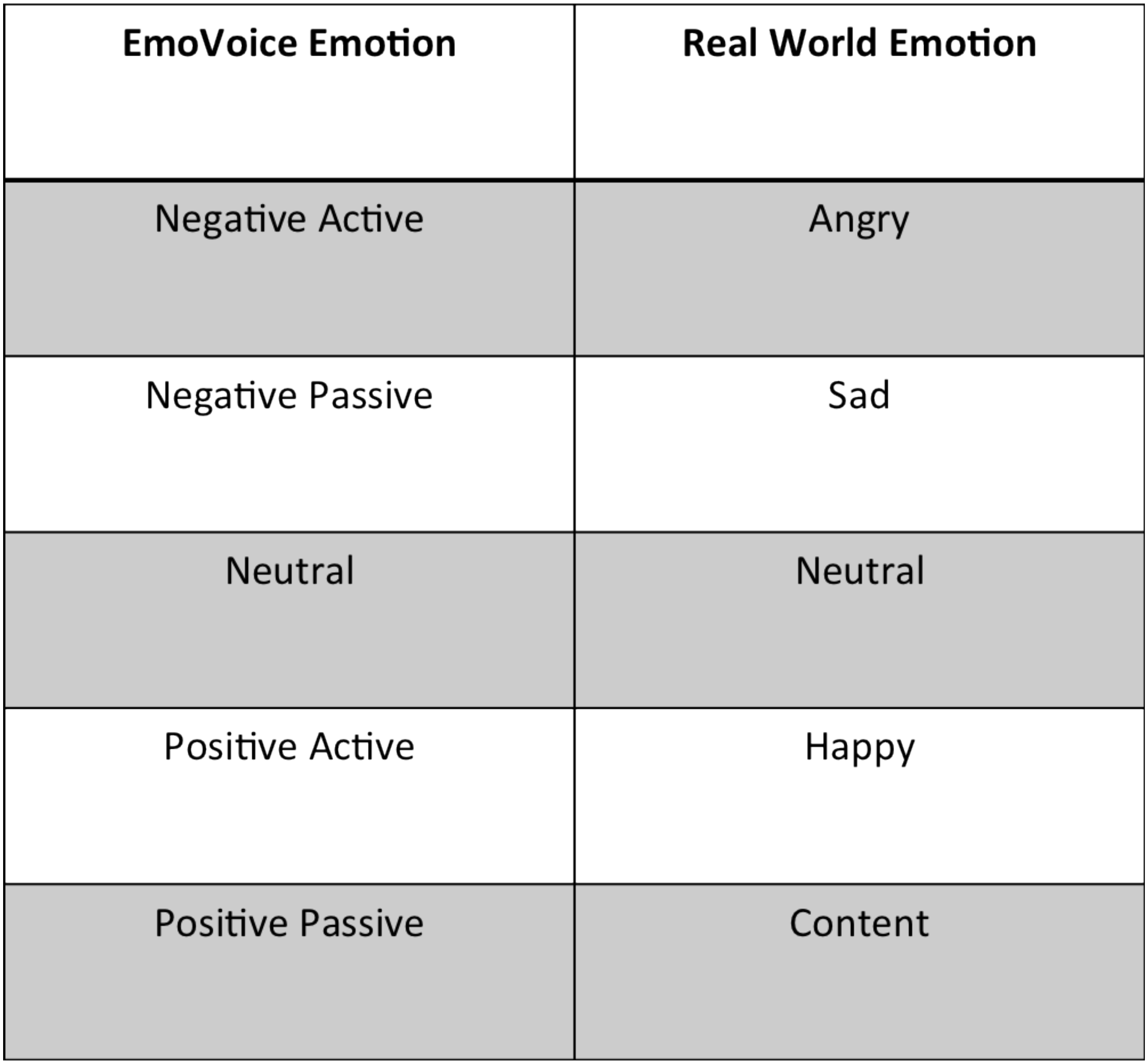
We decided to use two unimodal systems for our combination process, one which classifies emotion based on audio information only and another that works on visual information; more specifically, on still images of faces. In this section, we will discuss each piece of software and its functionality, as well as some unused software that was considered during research but was either scrapped or set aside for future work.

3.1 Audio System

The audio system chosen is called EmoVoice, an open source system that analyzes audio in real time and attempts to classify the emotional state of the speaker [12]. It was developed in the 2000s at the University of Augsburg, and has been used as the internal audio system for a variety of systems [5] [11]. Along with the source code, a collection of binaries is available on the project website for download and usage. The system uses a data model trained by a user reading a series of training sentences. Then, those training instances are used as the training data for a classifier. When activated, the system waits for input in real time. When input is provided via a microphone, the system performs its analysis and outputs to the screen. The output includes a classification of the user's emotion along with confidence levels [i.e. how confident was the system in its choice and the percentages of the unchosen options]. The main file is a Windows Batch file, which uses a few other files to specify the data model and other option files. The executable in its original state uses a default data model and classifies the user's spoken voice as either positive or negative. However, we wanted the system to classify between a wider range of emotions in order to give the audio system more weight in our analysis; a simple classification between positive and negative does not tell us much specifically. In the training process, the system uses a stimulus file, which contains a list of emotions to recognize, and training instances for each category [for example, the default stimulus file only listed positive and negative as possible classifications with training instances divided strictly between those two outputs]. The system included a wide range of stimulus files to choose from, which included a ”long” stimulus file. The long stimulus file introduced passive and active voice, along with a neutral state, into the list of possible outputs, with training instances for each new category. At this point, the system could classify between ”positive passive”, ”positive active”, ”negative passive”, ”negative active”, and ”neutral”. Some examples of stimuli sentences are listed below.

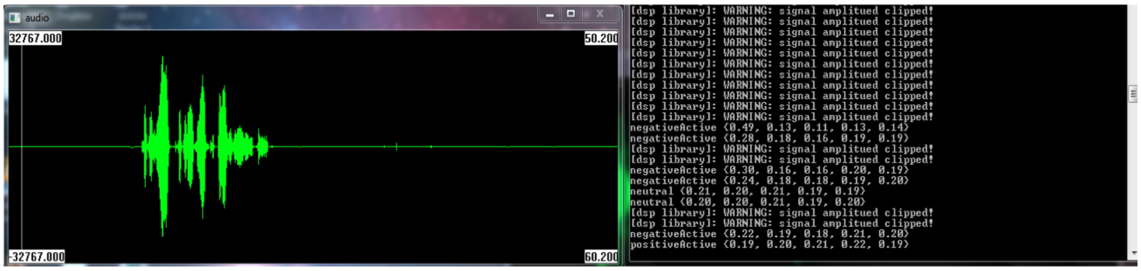
1. I'm in a good mood today. - ”positive passive”
2. Great news! I got the job! - ”positive active”
3. When I need you, you're never there! - ”negative active”
4. I can't seem to do anything right today. - ”negative passive”
5. There are two trains to Albany today. - ”neutral”

The direct relationship between the classifications of EmoVoice and real world emotions are shown in Figure 1.



EmoVoice Emotion	Real World Emotion
Negative Active	Angry
Negative Passive	Sad
Neutral	Neutral
Positive Active	Happy
Positive Passive	Content

Figure 1: A table showing the connections between the outputs of EmoVoice and the appropriate translation to their corresponding real world terms



```
[dsp library]: WARNING: signal amplitued clipped!
[dsp library]: WARNING: signal amplitued clipped!
[dsp library]: WARNING: signal amplitued clipped!
[dsp library]: WARNING: signal amplitued clipped!
[dsp library]: WARNING: signal amplitued clipped!
[dsp library]: WARNING: signal amplitued clipped!
[dsp library]: WARNING: signal amplitued clipped!
[dsp library]: WARNING: signal amplitued clipped!
[dsp library]: WARNING: signal amplitued clipped!
[dsp library]: WARNING: signal amplitued clipped!
[dsp library]: WARNING: signal amplitued clipped!
negativeActive <0.49, 0.13, 0.11, 0.13, 0.14>
negativeActive <0.28, 0.18, 0.16, 0.19, 0.19>
[dsp library]: WARNING: signal amplitued clipped!
[dsp library]: WARNING: signal amplitued clipped!
negativeActive <0.30, 0.16, 0.16, 0.20, 0.19>
negativeActive <0.24, 0.18, 0.18, 0.19, 0.20>
neutral <0.21, 0.20, 0.21, 0.19, 0.19>
neutral <0.20, 0.20, 0.21, 0.19, 0.20>
[dsp library]: WARNING: signal amplitued clipped!
[dsp library]: WARNING: signal amplitued clipped!
negativeActive <0.22, 0.19, 0.18, 0.21, 0.20>
positiveActive <0.19, 0.20, 0.21, 0.22, 0.19>
```

Figure 2: A snaphshot of EmoVoice in action

We trained a new model to replace the default model that came with the system, which also allowed us to fully implement the longer stimulus file containing not only the wider range of emotions, but a longer set of training instances. In addition to the main batch file, there was a file for training new models. The user simply has to read through the list of training instances, which are divided into subsets for each of the five emotions of the longer stimulus file, and then train a classifier on the audio instances provided through the user's speech. We used a provided Naive Bayes classifier and 5-fold cross validation to produce the appropriate files for the audio system to use. It is relatively easy to create new models and then change the software to use those new model files, although care must be taken that a particular model must be used on the same computer, using the same microphone, on which it was trained, since the hardware used in the collection of audio input can have an effect on the system's performance.

When introduced by its developers, a wide range of classification accuracies were recorded. The results of these initial studies ranged from 80 percent to as low as 41 percent. A study that introduced a Companion system, which would serve as a dialogue system to a user and offer input or feedback based on what it determined the user's emotional state to be, showed performance of 47 percent accuracy[2]. During the initial wave of experiments, we planned to produce an accuracy on our own dataset using EmoVoice.

3.2 Visual Software

The visual software was provided by Professer Shane Cotter at Union College. It uses Principal Component Analysis, in which local regions of the face are first examined and their information is combined at the end vs. looking at the face as a whole, and we used k-nearest-neighbors with the euclidean distance metric as the classifier [4]. K-nearest-neighbors classification works by taking a new, unclassified instance and comparing the classifications of its ”neighbors” [the existing instances that are most like the new instance] in order to classify the new instance. The ”K” refers to how many instances, or ”neighbors” are examined during the classification process. The software takes still images of faces as input and the software was initially trained on the Japanese American Female Facial Expression Database, which is available for public use [7]. There were only a few minor changes that had to be made, most of which were related to introducing new input into the system. The name of each image must be in a specific format, which includes two codes: one identifying the emotion that the user is displaying, and the other identifying who the user is. When loading a group of images, all images must have the same dimensions, and all pictures must be in greyscale. To introduce more users into the system simply means updating the software's internal list of users. The system inially wrote its performance percentages into a data file. For our purposes, we wanted to output the classifcations of the system to a text file, since the audio system can do the same and it would make the combination of the two systems much easier, so some small modifications were made to output the system's individual results to a text file. One interesting note to make about the software is that the user must manually click on the eyes for each image loaded, so the system knows where the eyes are located. Aside from making testing of large datasets more time consuming, this could have some interesting implications depending on how the multimodal system ends up being used. For example, there is potential for the multimodal system to be mounted on an existing robot at Union College, and we cannot perform the eye finding process manually when the software is being used by the robot itself. This is addressed further in the Future Work section. Multiple experiments were conducted with the visual system during its study involving removal of certain regions of the face. The highest published accuracy of the software was 93 percent, with some results on specific experiments lower than 60 percent.

3.3 Weka Machine Learning Software

For the final classification, we decided to use the Weka Machine Learning Software, which can run either via a GUI or via Java code. [1] The software contains many different classifiers and allows for various modifications to the dataset via attribute removal or filtering before running a classifier. The software also allows a user

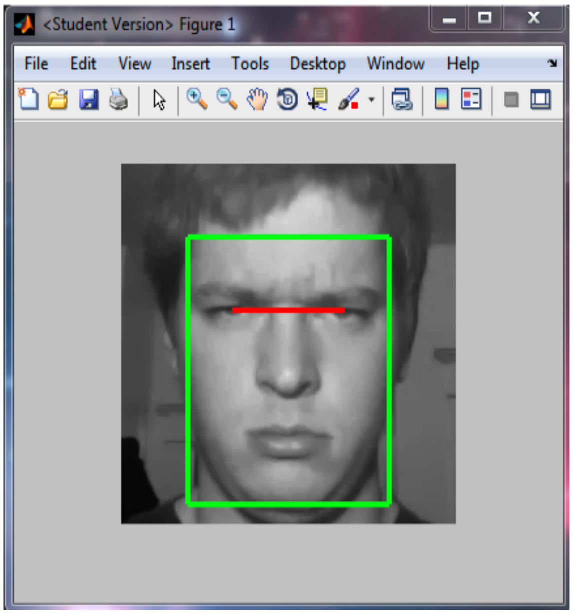


Figure 3: The visual software after identifying eye location

to run statistical significance tests. The software can load datasets written in various file formats; we will be using the Atribbute Relational File Format. The structure of the file defines the attributes that an instance of the dataset can have, what type [ex: numerical, nominal] each attribute is, and then the list of instances are provided, each with a class attribute [what category that particular instance falls into; in our case, it represents what emotional state the particular attribute belongs to]. When running a classifier over some dataset, by default, the software outputs the number of correctly classified instances along with the number of incorrectly identified instances.

3.4 Unused Software

A side goal of the study, which was initially discussed in the introduction, was the automation of the software, meaning that the system would take in audio and visual input simultaneously, run the unimodal systems on their individual inputs, then take the outputs of each system and create a new instance which could be run through a classifier. One potential form that this could have taken was via a simple GUI, implemented via the Java Swing library, which could gather input from the user and perform the necessary actions to produce a final classification. Over the course of the project, there was some software that was considered to either make the system more automated or connect the two unimodal systems efficiently, as we will layout in Section 5. Ultimately, the automation and connection process gave way to the research concerning the actual datasets, so there is some software that was experimented with and left unused, or that could still be re-considered in future work.

3.4.1 Alternatives to Visual Software

A great deal of exploration was done in the visual realm. Originally, we were looking for another open source recognizer that worked in the visual modality, since we had already acquired software for the audio modality at that point. There were some low level recognizers written in OpenCV, but none of the open source systems were at the level we desired them to be. For example, there are small portions of code, freely available, that can tell whether or not a user is smiling. This, like the initial classifcation state of EmoVoice [positive vs. negative] was not broad enough to suit our needs for a wider variety of possible outputs. Other software that classified facial images that performed with high accuracy were available only through license and payment, and only to other companies or official research groups. We were able to acquire Professor Shane Cotter's code at a relatively early point in the project so not too much time was lost finding suitable visual software.

3.4.2 Automation and Linking of Unimodal Software

During the first half of the winter trimester, we spent a good deal of time looking into system automation. We were aiming for a simple GUI and when input began, the unimodal systems would operate in sync and then deliver their output to the classifier. Most of this work focused on making the two unimodal systems run together. EmoVoice, as a batch file, was relatively straightforward to run inside Java as a command. Initially, we did look into some Java Piping before running it as a command inside Java, since we discovered we could make EmoVoice print its output to a text file, and so we would not have to worry about returning the audio input via a Pipe. Running the visual software from Java proved to be a little more complicated. This ultimately made us set aside the combination process and focus entirely on the datasets themselves. We explored an API called matlabcontrol, which is designed to connect to MATLAB from within a Java file and then execute MATLAB functions. However, we could not get this to work properly. We tried a second approach in which we used the MATLAB compiler, which the department has access to, to create shell script files. However, we could also not get these to run at first. While, with some tinkering, we could have undoubtedly made this all work, it was not the main focus of the project. Another issue that arose was the fact that EmoVoice is a Windows Batch file, intended only to run on that operating system, and we could only run all of the necessary visual software, which not only included Professor Cotter's code, but the code to snap and crop images, on the iMacs in the lab, since only those computers had access to all the necessary toolboxes. There is an API called Wine, which allows users to run Windows files on MAC OS X, but ultimately we decided that we could not spend any more time trying to combine the two systems and instead needed to focus the time we had left on building a suitable dataset and performing experimentation.

3.4.3 Online Classification

For online classification, or overall classification in real time, it is relatively straightforward to import and run classifiers into Java code, via the Weka external jar file. We are not sure how the classifiers will output within Java code; they will likely just produce an output similar to running the same classifier in Weka: number of instances correct, number incorrect, etc. If this is the case, we will need to figure out how we can list the classifications that the system came up with, so that when the classifier runs in real time, using a data file with a single instance of multimodal data, then a single, overall classification can be returned.

4 Data Gathering and Creation

In order to test the multimodal system, we needed audio and visual data from a number of users, and we also needed to associate both of the inputs provided by the user [i.e making sure that all of a particular user's audio input for "angry" was matched up with the same user's visual input for "angry", etc.]. In addition, we needed the dataset to train a classifier to categorize multimodal instances. This same classifier would be given audio data only and visual data only in order to compare the performance of a multimodal dataset to the performances of the unimodal datasets. In this section, we will discuss how we defined our dataset, how we gathered our data, and the preparations we made to the data in order to use it as input to the Weka software.

4.1 Defining and Collecting Data

The instances in our dataset would consist of the outputs from the two systems: EmoVoice's classification of the audio input, along with the confidence levels it produced, and the PCA software's classification of the visual input. The class emotions, which our final dataset would classify between, were "angry", "happy", "neutral", and "sad". There were two reasons for choosing these four emotions for the final classification: firrst, these emotions would be the easiest for the test users to understand and mimic for the purposes of visual data gathering. This also held true for EmoVoice, since "content", the only emotion of EmoVoice that didn't directly line up with one of the visual class emotions, was also relatively easy for people to understand. Secondly, these four classes were the emotions that the two unimodal systems had in common, and so we could generate data from each system that would match up evenly with output from the other. For our data, we gathered both audio and visual data was gathered in the same sitting. Our initial user set consists of eight people: five male and three female. The visual data was gathered in the following manner. After calibration, we ran a script that snapshotted the user several times, cropping and greyscaling the image in order to format them for use in the visual software, and writing them to file. For each series of images generated, the user was asked to express the four class emotions that we would be classifying between. The initial images gathered were generated with the users sitting at regular distance from the monitor, which was about two feet. After regular distance images were gathered, the users were moved back to a distance of roughly six feet. The reason we gathered this second round of visual data was to see how the visual software would react to a degradation in image quality; the idea was that moving away from the camera would result in a lower resolution face and make features harder to detect and classify. Also, there are plans to mount a final system onto an already existing robot, and since people talking to the robot would likely be talking to it at a similar distance that one might talk to a human when approaching them, at about 6 feet or so, we wanted to gather data that could indicate what performance might be like when using these methods on the robot itself. After the visual data was gathered, the user was asked to read a series of sentences to EmoVoice. The test sentences were divided into five subsets: "angry", "sad", "neutral", "happy", and "content".

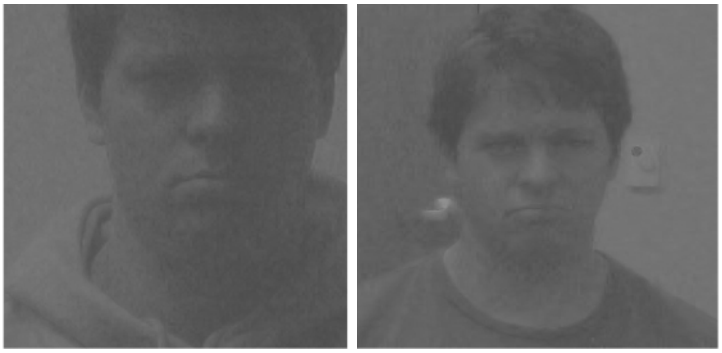


Figure 4: A sample of long vs. regular distance visual data

4.2 Authentic Emotion and Long Distance Audio Data

This two-step process of data gathering is not entirely authentic, as a system running in real time would be gathering audio and visual data simultaneously, as humans do. However, speaking alters your face, and a large amount of information could be lost from the visual realm, especially if we used images that made extreme deviations from the emotion that the user is truly trying to express. There have been some studies conducted on gathering authentic data, which explores some interesting techniques for elimintaing bias [10]. However, this was not the point of our study, although we did discuss some potential alternative experiments in which we did not tell the user exactly what the data gathering was for. This possibility still remains on the table for future work, but for the purposes of our initial data gathering, we did not focus on attempting to deceive the user. We wanted to get the best data possible for each system, and so this is why we conducted data gathering in this manner. No long distance audio data was gathered. We are unsure if this will have any major effects on the classification accuracy of EmoVoice on our data; if the input quality is the same using the long distance microphone as it is using the short distance microphone, and we retrain the data model using the long distance microphone, it is not unrealistic to expect similar classification accuracy.

4.3 Preparing the Data

To organize our data and run classifiers, as well as statistical tests on the results, over the data, we decided to use the Weka Data Mining Software [1]. It is a freely downloadable piece of software that allows the user to easily load and modify data via filters or manual modification. The software also comes outfitted with a wide variety of classifiers, as well as other tools such as statistical testing. There are also capabilities built-in to build datasets and run classifiers from a Java file by importing the Weka .jar file. Since we would be using this software, we need to edit the data into a format that could be compatible with that software. The Attribute Relation File Format, .arff, was chosen for this study. Each system has the capability to write its output into a text file. We wrote code, that will eventually fit into the combination area of the system layout, that takes two text files as input, creates a new .arff file and sets up the format for the file, including the attribute names, types, and possible inputs, and then used the text files to fill the new .arff file with instances. This made it relatively easy to generate datasets after gathering new data. For each series of experiments that we ran, three datasets would be generated. First, we would create the .arff file as just described, which represented the multimodal dataset. Weka makes it incredibly simple to remove or edit data, as well as apply filters quickly, and then save the new version. We would generate audio data .arff, which was the multimodal data with the visual data removed, and the visual data .arff, with the audio data removed.

5 Overall Design

The overall system design is shown in Figure 5. The layout of the system is relatively straightforward. The user will speak into a microphone while showing an expression to a camera. The audio input will be analyzed in EmoVoice in real time and the output will be written to a file. The visual input will be formatted to fit the requirements of the visual software. The visual software will analyze the image after the user has identified the location of the eyes, and the output from the visual system will also be written to file. The output of the audio system, which includes the classification of EmoVoice as wel as the outputted confidence levels, will be combined with the visual output, which consists of only the visual software's classification, will be combined together into an appropriate formatted file and then run through a classifier, which wil produce a final classification of the user's emotional state. Ultimately, we want to get to a point where, once the input is iniated, then the entire process will become automated, or at least semi-automated.

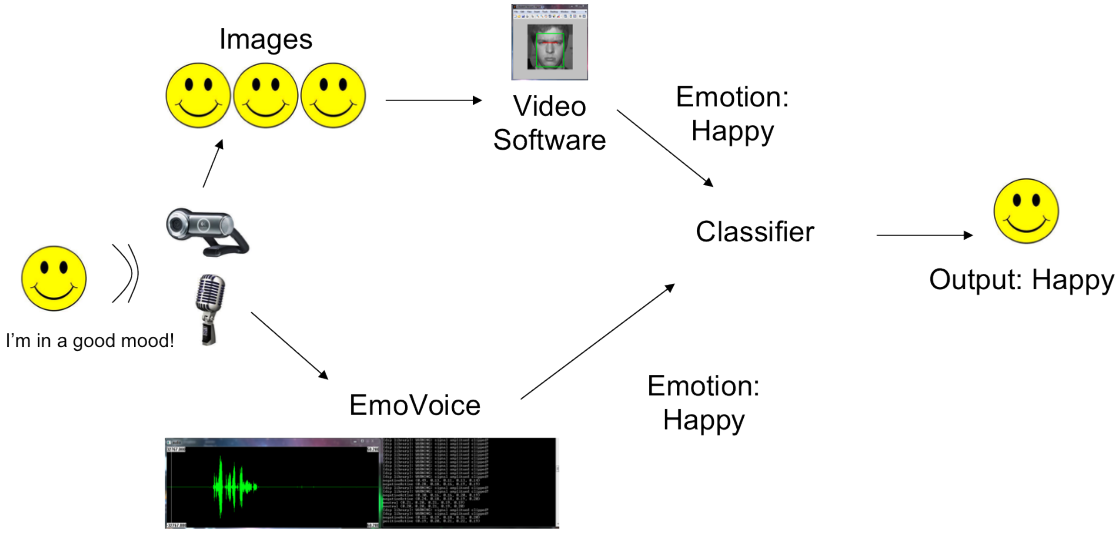


Figure 5: The overall layout of the system

6 Initial Experiments

The final dataset consisted of 288 training instances. Initially, two experiments were run: one using the long distance visual data, in both the visual dataset and the multimodal dataset, and one consisting of regular distance data. For each experiment, the three datasets [audio only, visual only, multimodal] were run through the same classifier while conducting a t-test to determine statistical significance of performance. Instead of testing a multitude of classifiers, the J48 classifier was chosen. This classifier implements a decision tree via the C4.5 algorithm [13]. A decision tree is a structure that will determine an output based on

1. The structure of the tree. Each node contains a certain decision and each branch represents the path taken by the outcome of the previous decision.

2. The input of a new instance.

This classifier was chosen mostly due to our familiarity with its functioning and it would be easy to examine the decision trees produced by each dataset, which would allow us to see how the multimodal dataset was using each modality. An example showing the top of decision tree for our initial multimodal dataset is shown in Figure 6. All calculations made via a decision tree start at the root node. Each line from the root node is labeled with the decision that was made to follow that branch. For example, when first examining the new instance, if the PCA Class output was Angry, then the leftmost branch from the root node was followed, and then after reaching Node 2, a second decision was made based on the attributes of that instance. In this case, if EmoVoice's classification for that input was "sad" or "content", then the overall classification of the instance would be "happy". If the classifcation of EmoVoice, for that instance, was "happy", "neutral", or "angry", then we would move further down the tree.

The results of the initial experiments are show in Figure 7. The performance of the audio and visual system on our dataset was lower than the published accuracies we introduced earlier, but these differences can be explained. For the audio system, if the features of the user's voice deviate far from the features of the voice that trained that model, performance can drop. Our audio model was trained using a male voice and we had some female users in our initial dataset. While the accuracy varied between each user, the accuracy of EmoVoice took a clear drop when classifying the inputs from our female users. The visual system was initially trained on faces of Japanese American Females [7], and while we had female users in our dataset, a majority of users were male, and none of our users were Asian in ethnicity, and much like the audio modality, the visual performance dropped when introduced with unfamiliar data. There was some debate on whether to retrain the visual software using our own data, but the performance did not drop low enough to warrant this. The highest published accuracy for the visual system is 93 percent, while we were experiencing an accuracy of 77 percent. While there was a decrease in accuracy, the classification rate was still fairly high and the drop was not drastic enough to warrant retraining the visual software.

It is clear that the intial combination of the two modalities, with the data produced by each unimodal system untouched, was ineffective. Only EmoVoice performed statistically worse than the multimodal dataset, which is not very interesting due to its already low performance. The multimodal dataset and visual only dataset performed statistically at the same level. This was clearly due to the extremely poor performance of the audio data. Looking at the decision tree shown in Figure 6 also reveals how reliant the multimodal dataset was on the visual data. The initial decision only takes the visual data into account, and then, based on that original decision, some of the audio data is referenced to make the later decisions. However, for visual data classified as Happy, the overall classification is immediately labeled as Happy without even considering

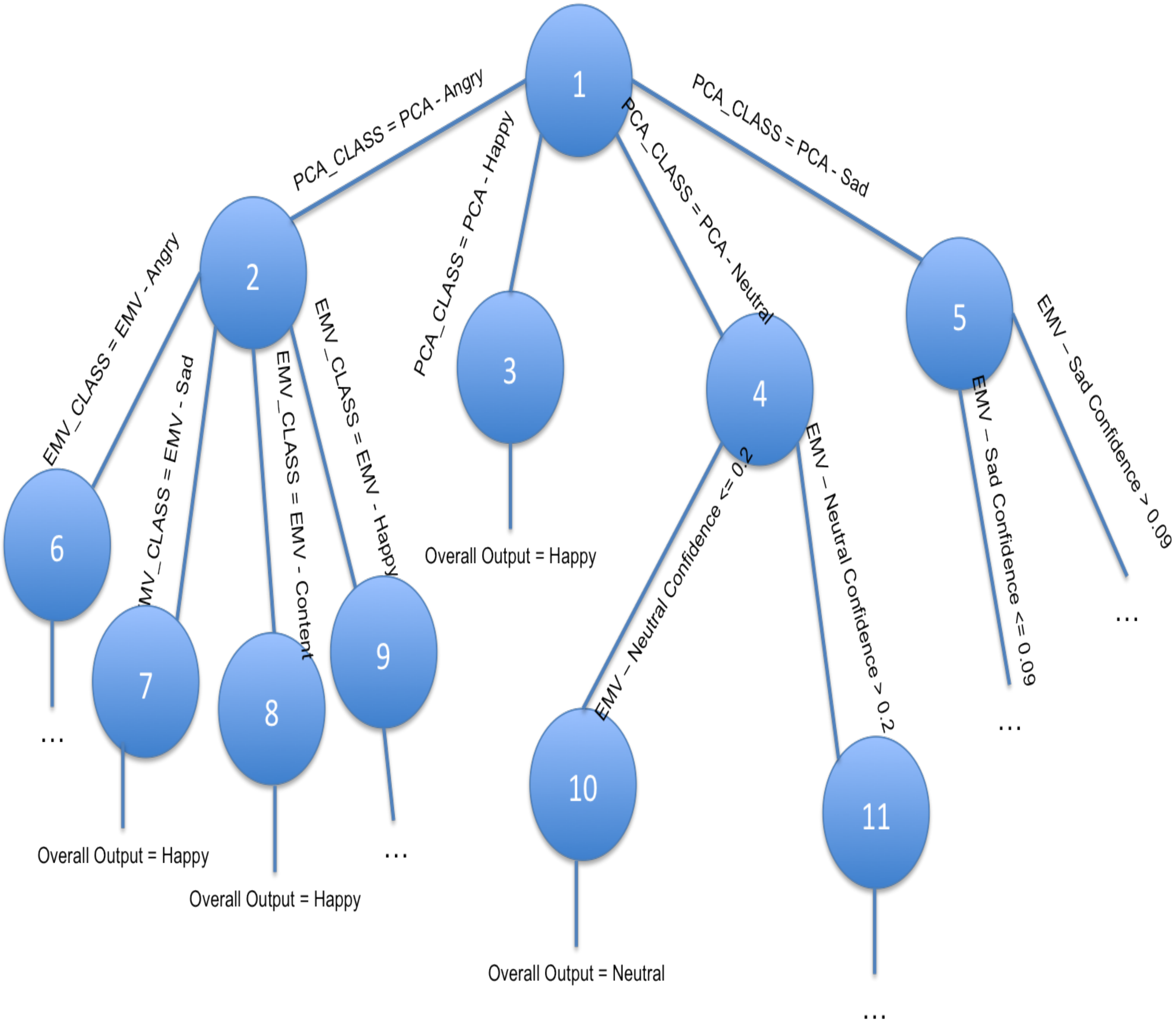
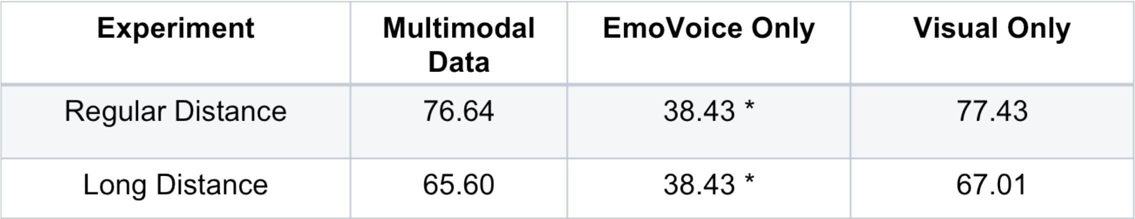


Figure 6: The top of the tree generated by C4.5 using the initial multimodal dataset



Experiment	Multimodal Data	EmoVoice Only	Visual Only
Regular Distance	76.64	38.43 *	77.43
Long Distance	65.60	38.43 *	67.01

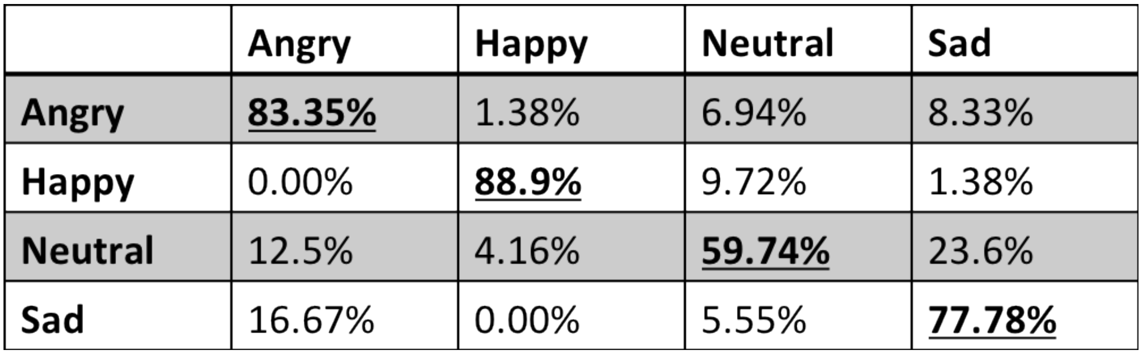
Figure 7: Initial experiments sorted by dataset. A * indicates that the corresponding result was worse than any other result in the same row that does not have a *

the audio data. In some cases where both the audio and visual data agree, there is even further analysis of the audio confidence levels that takes place. This is not a good example of using more audio data; in fact, it shows the inaccuracy of EmoVoice on our dataset, since, in an ideal situation, if both of the outputs from the unimodal systems agreed and both systems were fairly accurate overall, then hopefully the classifier would immediately choose the agreed-upon emotion. EmoVoice's accuracy at this point was extremely low, and so the decision tree could not completely rely on the classifications generated by the audio system.

7 Manual Bias

After the initial experimentation phase, it was clear that the simple combination of the audio and visual data, due to the high performance of the visual software and the unrelaibility of the audio software, was ineffective over the visual software alone. We needed to refine our combination process in order to achieve higher performance afterwards. Previous research [6] has shown that different modalities have tended to complement each other. Working with our systems and examining ther invidual performances on our dataset allowed us to identify the strengths and weaknesses of the two systems. Ultimately, we wanted to identify two pieces of information:

1. How could we go about improving the visual software? What emotions does the visual software perform poorly on?
2. What biases does EmoVoice have that result in its poor performance?



	Angry	**Happy**	**Neutral**	**Sad**
Angry	**83.35%**	1.38%	6.94%	8.33%
Happy	0.00%	**88.9%**	9.72%	1.38%
Neutral	12.5%	4.16%	**59.74%**	23.6%
Sad	16.67%	0.00%	5.55%	**77.78%**

Figure 8: Confusion matrix for regular distance visual data

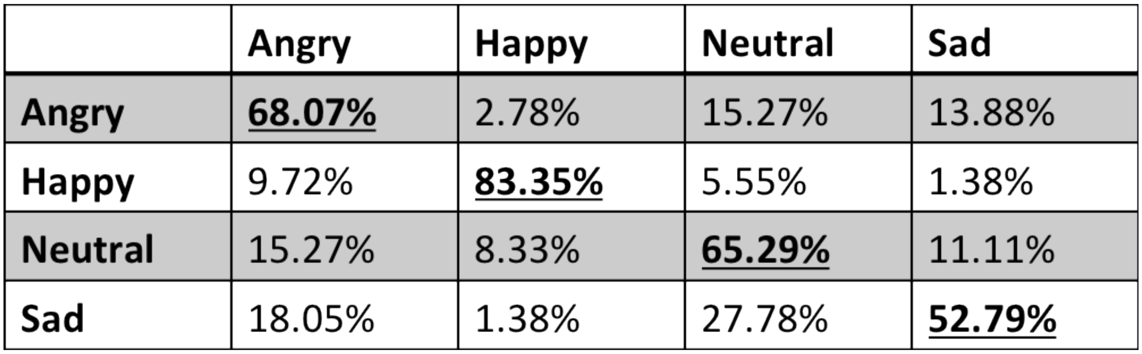
We wanted to not only improve the audio software's performance so that its contribution to the multimodal dataset will be reliable, but improve EmoVoice in a way that both targeted its weaknesses and those of the visual software, so that where the visual software was missing, the audio data could make up for that loss and ultimately give the multimodal dataset a higher classification accuracy.

7.1 Deficiencies of the Unimodal Systems

We examined the performance of the visual data to identify its weaknesses. The confusion matrices of the visual dataset, using regular distance data, for each emotion are shown in Figure 8.

It is clear that, while the visual software had a decent performance overall, there are several areas of weaknesses. The system was least accurate on "neutral" faces, misidentifying almost half of the training instances. "Sad" was more or less performing at the overall level of accuracy for regular distance faces, but this is still almost a quarter of instances misidentified. "Happy" and "angry" were extremely accurate, but the visual software was still making mistakes for those two emotions. In short, the classifier was experiencing trouble across all emotions. This becomes even more clear when looking at the performances for long distance data, show in Figure 9.

When supplied with degraded data, the accuracy of the visual software dropped even lower. Except for "neutral", all of the emotions experienced a drop in classification, with accuracy on "sad" facial expressions dropping by about 25 percentage points, and "angry" dropping by about 15 percentage points. For both regular and long distance data, the software was making mistakes across all instances, and "sad" and "neutral"



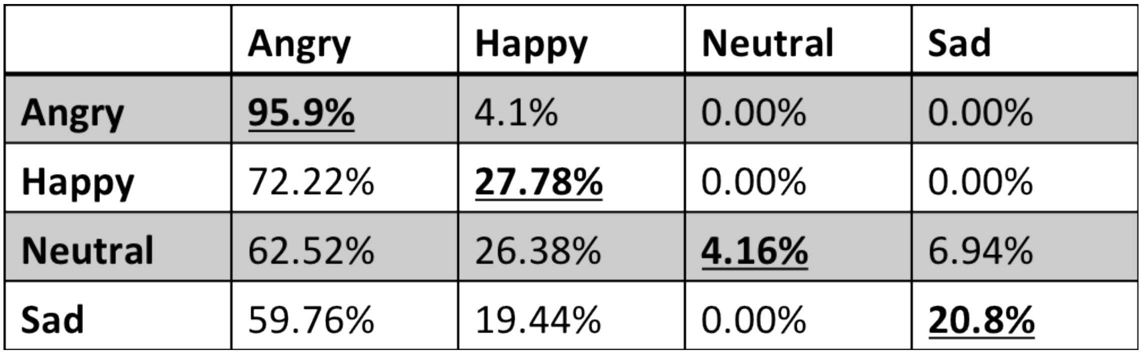
	Angry	Happy	Neutral	Sad
Angry	**68.07%**	2.78%	15.27%	13.88%
Happy	9.72%	**83.35%**	5.55%	1.38%
Neutral	15.27%	8.33%	**65.29%**	11.11%
Sad	18.05%	1.38%	27.78%	**52.79%**

Figure 9: Confusion matrix for long distance visual data

the least accurate for both systems.

This made it easier for us to improve EmoVoice's accuracy since we could make improvements to EmoVoice for all emotions. Working with EmoVoice and examining its performance reveals two biases: a strong bias towards the negative voice, Angry and Sad, and an extremely strong bias towards the active voice. This leads to its low accuracies on the positive emotions, as well as its poor performance on Sad, since the system classifies the emotion as negative but the strong active bias leads the system towards Angry rather than the correct classification. Both of these biases combine to result in a low performance on Neutral. These biases are made clear when examing the confusion matrices for the audio dataset, show in Figure 10. For "happy", "neutral", and "sad", we can see that 72.22, 65.52, and 59.76 percent, respectively, of the instances are misclassified as "angry". For the active bias, we can see that 59.76 percent of "sad" instances are misidentified as "angry" and 19.44 percent are mistaken for "happy" ["happy" is equivalent to "positive negative"; while there is negative bias in EmoVoice, it is outweighed by the bias towards the active voice]. "Neutral" is heavily affected by the 65.52 percent misidentified as "angry" and the 26.38 percent mistaken for "happy", showing both the active and negative bias of EmoVoice.

"Angry" is extremely accurate due to the biases, and dominated the rest of the emotions. For all other emotions, over half of the instances were misidentified with "angry" alone. "Happy" performed the 2nd best, demonstrating the active bias of EmoVoice. The system was only correctly identified "sad" about 1/5 of the time, and "neutral" was almost never correctly identified.



	Angry	Happy	Neutral	Sad
Angry	**95.9%**	4.1%	0.00%	0.00%
Happy	72.22%	**27.78%**	0.00%	0.00%
Neutral	62.52%	26.38%	**4.16%**	6.94%
Sad	59.76%	19.44%	0.00%	**20.8%**

Figure 10: Confusion matrix for audio data

7.2 Rule Creation

Knowing these weaknesses, we created a set of rules that we referred to as manual bias. For each subset of overall emotions (”angry”, ”happy”, ”sad”, ”neutral”), we would look at EmoVoice’s output for a particular instance. If the output was incorrect; that is, it was not identical with the overall classification for the training instance, then, depending on the overall classification, we would apply a particular rule to EmoVoice’s class output, and if that output aligned with the rule we created, then we would change the class output of EmoVoice to the correct emotion. We did not make any rules for the ”angry” training instances since EmoVoice had a high classification accuracy on ”angry”. The rules were as follows:

”Happy”: If the confidence levels of ”content” and ”happy”, added together outweighed all three remaining confidence levels individually, then change the EmoVoice class to ”happy”

”Sad”: If the confidence level of ”sad” was 2nd to the confidence level of ”angry”, and within 0.05 percentage points, then change the EmoVoice class to ”sad”

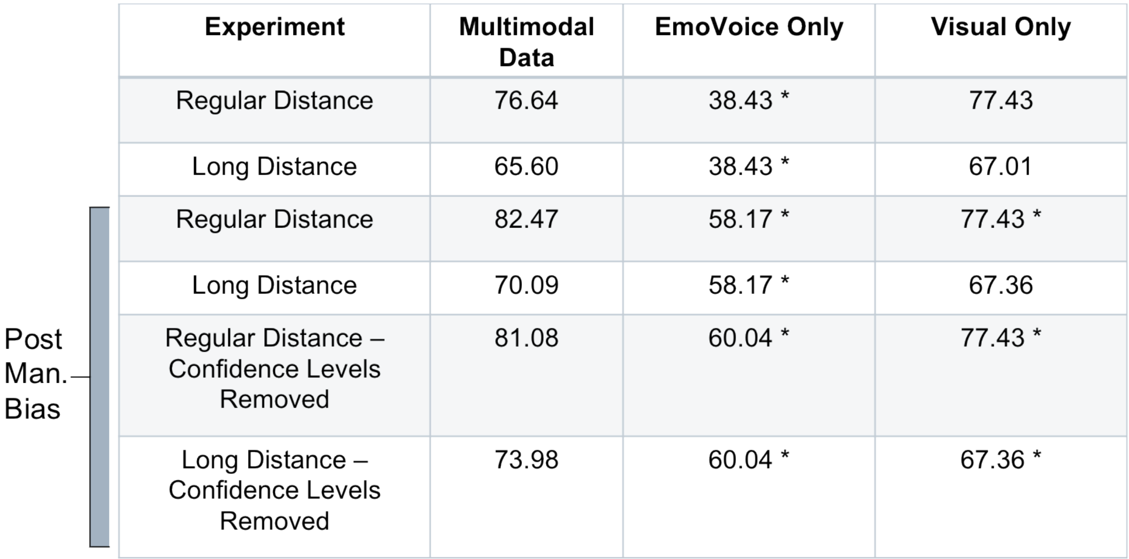
”Neutral”: First, if the confidence level of ”neutral” was tied with another confidence level, but EmoVoice had chosen the other class over ”neutral”, then change the EmoVoice class to ”neutral”.

”Angry” instances were ignored due to the strong negative and active bias present in EmoVoice. For ”happy” instances, if the confidence levels of EmoVoice for ”content” and ”happy” added together outweighed all other confidences, the classification of EmoVoice would be changed to ”happy” to combat the negative bias. If the ”sad” confidence level was 2nd to ”angry”, meaning that EmoVoice came close to correctly identifying the instance as ”sad” but the active bias intervened, the classification of EmoVoice would be changed to ”sad”. For ”neutral”, there were two cases in which there were multiple misidentifications. There were many instances in which the confidence level of ”neutral” was tied with another emotion, usually ”angry”, for the highest confidence level but EmoVoice had selected the other emotion due to the biases. Another scenario in which the EmoVoice classification would be changed to ”neutral” would be if all confidence levels were within a very close range of each other, meaning that the system wasn't too confident about any emotion, or that it had good reasoning to think that it could have been any of the five options. We considered this to be a ”neutral” state since EmoVoice had not strongly identified the user as being in one emotional category. After we applied these rules manually to the dataset, we were prepared to run the next set of experiments.

8 Experiments Using Manual Bias

After the application of the manual bias, four more experiments were run. We repeated the two experiments using the regular and long distance visual data, and then ran two more experiments, both with regular and long distance visual data, with the confidence levels of EmoVoice removed from the dataset. We would use the confidence levels of EmoVoice to apply the manual bias to the EmoVoice class attribute, and then remove them. The motivation for this subset was to see what kind of effect that would have on the audio system's performance and if it would have any major effect on the multimodal dataset. The results of these four experiments are show in Figure 11.

For three out of the four experiments, we saw that the multimodal dataset performed statistically better than the audio and visual data alone. The highest performance of the multimodal dataset was the regular distance with confidence levels. The multimodal dataset, for that experiment, had an accuracy of 82.47 percent, which was higher, with statistical significance p greater than 0.05, than the audio and visual performance of 58.17 and 77.43 percent, respectively. For regular distance, regular distance with confidence levels removed, and long distance confidence levels removed, we saw performances of 82.47, 81.08, and 73.98 percent, respectively. For long distance visual data, with the audio confidence levels, the performance of the multimodal dataset was higher than the visual dataset, but the difference was not statistically significant. It is clear that the manual biases we applied resulted in an increase of about 20 percentage points for EmoVoice,

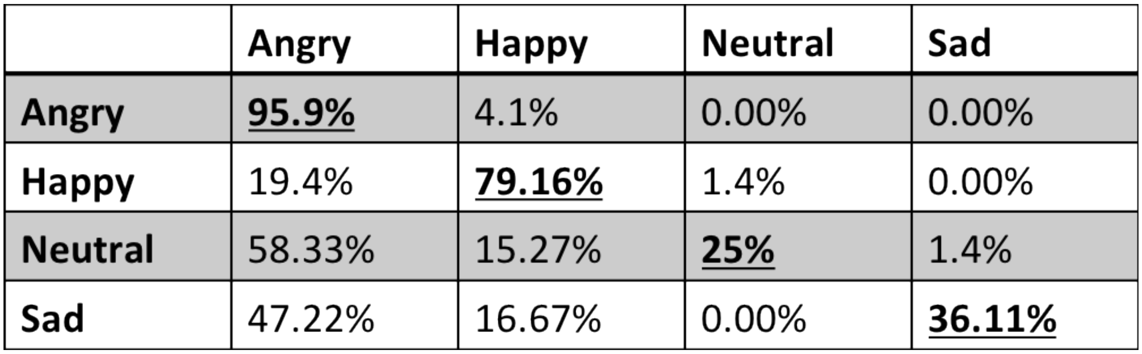


	Experiment	Multimodal Data	EmoVoice Only	Visual Only
	Regular Distance	76.64	38.43 *	77.43
	Long Distance	65.60	38.43 *	67.01
Post Man. Bias	Regular Distance	82.47	58.17 *	77.43 *
	Long Distance	70.09	58.17 *	67.36
	Regular Distance – Confidence Levels Removed	81.08	60.04 *	77.43 *
	Long Distance – Confidence Levels Removed	73.98	60.04 *	67.36 *

Figure 11: Experiments, conducted after manual bias was applied, sorted by dataset. The * indicates that the performance of a dataset for a particular experiment was statistically worse than any other dataset in that experiment that did not have a *

from 38.43 percent to 58.17 percent. and since the visual data was untouched, we can see that these rules are responsible for the increase in accuracy of the multimodal dataset. Another interesting pair of the results was that the removal of confidence levels increased the performance of the audio data by a small amount, from 70.09 percent to 73.98 percent. The performance of the multimodal dataset using the audio data minus the confidence levels remained statistically the same, but we saw an increase in multimodal performance when using the long distance data combined with the removal of the audio confidence levels. The effectiveness of the rules in raising the accuracy of EmoVoice for their particular emotion is demonstrated in Figure 12.

The performance of ”happy” almost tripled, and both ”neutral” and ”sad” also saw healthy increases in performance. We can clearly see that the misidentifications of ”happy” instances as ”angry” plummeted, although it was still occuring about 1/5 of the time. ”Neutral” reduced misidentifications across all other emotions, and the performance increase in identifying ”sad” removed many misclassifications of ”angry”, which was the intention, since we desired to combat the active bias present in EmoVoice, and also reduced the misclassifications with ”happy” by a few instances.



	Angry	Happy	Neutral	Sad
Angry	**95.9%**	4.1%	0.00%	0.00%
Happy	19.4%	**79.16%**	1.4%	0.00%
Neutral	58.33%	15.27%	**25%**	1.4%
Sad	47.22%	16.67%	0.00%	**36.11%**

Figure 12: Confusion matrix of audio data post manual bias

9 Future Work

We will be continuing this project into the spring trimester as a practicum, the end goal of which is getting the software to the point where it can be mounted on the robot in CROCHET [Collaborative RObotics and Computer-Human Emprical Testing] lab at Union College, or, barring that, get it as close as possible, where someone else can pick up afterwards. There are numerous issues that need to be addressed before we can consider the software to be ready for integration with the robot.

9.1 Use of the Visual Software on a Robot

First, there is an issue with the visual software in that it is written in MATLAB, and we do not currently have the ability to run MATLAB on Linux, which powers the robot. A simple port, as in converting the visual software to another language, would not be effective as the visual software makes uses of certain toolkits in MATLAB. Even though we have access to the MATLAB compiler, which can create shell script files from .m files, those script files are still carrying the MATLAB code with them. One possibe solution might be attaching a Mac Mini somewhere onto the robot and port the visual processing out to that computer, and return its result, as suggested by another faculty member John Rieffel. The issue has fixes, so it will likely not become a huge problem.

However, a second issue exists with the visual software: when the images are loaded into the software, the eyes must be clicked by the user manually, so that the software can further crop down the face and become aware of the eye location for appropriate feature extraction. To still use the video software, we may have to look into eye finding software, and this might also affect how we integrate the software onto the robot. Simply assuming that the user's face will be in the center of the robot's view, which would mean that their eyes are always in a particular location, would not be an accurate assumption to make.

9.2 Software Compaction and Manual Bias Refinement

Third, some more software must be written and combined together in order to compact the system. While we have some code written for different parts of the layout, and have an idea of how to implement other sections, such as calling the J48 classifier directly from the code using a new dataset generated on the fly, and calling the unimodal systems remotely from within a main piece of software, we still have some amount of software creation, editing, and combination ahead of us.

We also want to take another look at the manual rules and see if we can refine them, since they are not completely streamlined, so to speak. In particular, the rule for modifying the "happy" instances is extremely biased; for example, applying that rule to the whole dataset results in a majority of instances, no matter the emotion, switching to happy. We may look at somehow combining all of the rules so that they will be effective on the particular instances we want without explicitly applying the rule to only the instances we want it to apply to. Ultimately, we want the rules to be accurate enough that the classifier, when being trained on the dataset, will learn those exact rules. While the classifier currently learns the data that we give it as input, it does not explicitly know exactly what we did. If we can make the classifier aware of the biases present in both systems, then it can handle them when classifying new input and so we will not have to perform any modification on the data. Until that point is reached, we could still write the manual rules into the software, so that the data would still be modified before classification, but that process would become automated.

Finally, another clear requirement is testing the audio system using a microphone that can pick up audio at long distance, since the person that the robot interacts with will be at a longer distance. Retraining EmoVoice on this type of microphone would be straightforward, but we would need to gather more data in order to see if performance of the audio system is affected, and how the multimodal data will react.

Gathering more test subjects during this regathering process would be a nice touch. Overall, there are numerous interesting opportunities for improvement as well as considerations to make before mounting the software on the robot itself, so the practicum should produce some more interesting results.

9.3 Additional Modalities

Another potential addition to the system would be the analysis of gesture information in relation to emotion. There is some gesture recognition research ongoing at Union College, but not related to emotion recognition. A good deal of work would be involved in adding a new modality [what emotions would we use gesture information for, what would constitute a suitable gesture for each emotion, etc], and we would have to perform some more research on previous emotion recognition studies that used gesture information. Overall, it is an interesting possibility to consider.

10 Conclusion

In this study, we acquired two existing classifiers for two separate modalities: audio and visual. We developed a framework for combining the two systems, and created a new dataset which combined the outputs of each unimodal system in order to test for accuracy improvement. In addition, we gathered both long and regular distance data to test how the added factor of distance would play a role in visual accuracy. Initially, the simple combination of the two outputs was not a statistical improvement, since the accuracy of the audio system was very low and the visual system was fairly accurate. We examined the weaknesses and strengths of the two systems in order to see how we could combine them in such a way that we could exploit their complementary information. After developing a set of rules to improve the accuracy of the audio system where the visual system was failing, we saw a statistical improvement, using the multimodal dataset vs. using either of the unimodal datasets alone, for a majority of our experiments. The highest accuracy of the multimodal dataset was 82.47 percent, which was a statistical improvement over the audio and visual accuracies of 58.17 and 77.43 percent. Potential future works includes further automation and refinement of the system and the possibility of mounting the system onto a robot.

References

[1] Machine Learning Group at the University of Waikato. Weka 3: Data mining software in java.

This software has an assortment of classifiers as well as techniques for modification and filtering of data before applying any training. It also provides the ability to run statistical significance tests, which made it easy for us to compare performance between multiple datasets.

[2] David Benyon, Jay Bradley, Bjoern Gambaeck, Preben Hansen, Oli Mival, and Nick Webb. Deliverable d.1.4.3 - companion's evaluation results. Technical report, 2011.

This is a report from the Companion system, showing some test results. The table showing the performance of EmoVoice is the data were are interested in for this citation.

[3] George Caridakis and Givevra Castellano. Multimodal emotion recognition from expressive faces, body gestures, and speech. In *Artificial Intelligence and Innovations 2007: From Theory to Applications*, volume 247/2007, pages 375–388, 2007.

This paper was another interseting look at multimodal recognition as it also included discussion of a third modality, gesture. The study also used a simple set of rules for decision classification, and it was found that this was not as effective as feature level (while only using a set of rules).

[4] Shane Cotter. Recognition of occluded facial expressions using a fusion of localized sparse representation classifiers. In *Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop, 2011 IEEE*, pages 437–442, 2011.

This paper introduces the software for recognizing emotions of still images of facial expressions.

[5] F de Rosis and C Pelachaud. From greta's mind to her face: modelling the dynamics of affective states in a conversational embodied agent. In *International Journal of Human-Computer Studies*, volume 59, pages 81–118, 2003.

This is a citation of one system that implemented EmoVoice as part of a larger program, in this case, a conversational agent.

[6] Zhigang Deng and Carlos Busso. Analysis of emotion recognition using facial expressions, speech, and multimodal information. In *Proceedings of the 6th international conferencee on Multimodal Interfaces*, pages 205–211, 2004.

This paper provided a good first overview of how multimodal fusion was performed for emotion recognition for audio and visual information, and also introduced feature level and decision level fusion.

[7] Miyuki Kamachi. The japanese female facial expression (jaffe) database.

This is the database of images of various facial expressions used by Shane Cotter in his research on occluded facial expressions. This database is freely downloadable. - http://www.kasrl.org/jaffe.html

[8] Chul Min Lee. Toward detecting emotions in spoken dialogs. In *IEEE Transactions on Speech and Audio Processing*, volume 13, pages 293–303, 2005.

This paper stuck out from the others because their study attempted to analyze more than just acoustic information (lexical and discourse) in order to classify emotions for several reasons (ex: finding that certain words were often associated with a particular emotion). Their study showed improved performance when combining other information categories. It is certainly interesting, but I am not sure if I have the time to look at more than acoustic signals.

[9] Valery A. Petrushin. Emotion in speech: Recognition and application to call centers. In *In Engr*, pages 7–10, 1999.

This article discussed experiments in which people's ability to judge certain types of emotions were gauged, as well as specific aspects of the spoken word that they deemed most important to recognizing certain emotions. It was found that certain emotions were easier to recognize than others. These aspects of speech that were found to be important were used to train neural networks. The article also talked about applications to a call center in which a caller's emotional state could be classified.

[10] Nicu Sebe and Michael S. Lew. Authentic facial expression analysis. In *Image and Vision Computing*, volume 25, pages 1856–1863, 2007.

While not about multimodal fusion, this paper discusses some of the issues with gathering emotion data in a forced setting, and presents an interesting technique for gathering authentic data that could provide some inspiration for a real-life experiment of our own

[11] Jana Sichert. Visualisation of emotional expressions in voice. *VDM Verlag Dr. Mueller*, 2008.

This is another citation of a system that uses EmoVoice as part of a color kaleidoscope system that reflects the user's emotional state.

[12] Elisabeth Andre Thurid Vogt and Nikolaus Bee. Emovoice - a framework for online recognition of emotions from voice. In *Perception in Multimodal Dialogue Systems- 4th IEEE Tutorial and Research Workshop on Perception and Interactive Technologies for Speech-Based Systems*, volume 5078, 2008.

This paper introduces an online emotion recognition system called EmoVoice. The article describes how the system works, and shows several examples of EmoVoice implemented in other applications.

[13] Frank Witten and Hall. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan-Kaufmann, 2011.

This book provided some explanation to the functioning of decision trees, and thereby our classifier, C4.5