Final Project Review

SDP Team 23
SDP Team 23

Liam Shea

Jason Nguyen

Phillip Ashe

Paul Mahoney
Why Voice Biometrics?

● Benefits of Voice Biometrics
  ○ Easier to use than e.g. iris scanner
  ○ Low cost (no specialty hardware needed)
  ○ Can be used as an added layer of security in many applications
Project Goals

Create a reliable, portable system that could provide near real-time speaker identification.

Should exist as a pair of platforms:

- A mobile platform that records and process voice signals, and output identification features
- A server platform to store to the features and match features against those stored in database

New goals after drawbacks:

- Software working properly with at least 75% accuracy
- Entire system runs on a PC
- Program runs entirely in MATLAB
Original Block Diagram

- **Microphone**
  - Audio input
  - Lowpass Filter
  - Preamplifier
  - Power Amplifier

- **Raspberry PI**
  - Create User/Train Classifier
  - Database
  - Verify User via Classifier
  - Chose Mode Main()

- **Outputs**
  - LCD
  - Keyboard
Current (software) block diagram

- Input
  - Speech
  - Mode Selection

- Preprocessing
  - Features Extraction

- Model Training
  - Synthesize Data Set
  - SVM Algorithm

- Model Database

- Matching Modes
  - Verification
  - Identification

- Decision
System Overview

User starts the program in MATLAB

User is prompted to choose one of three modes of operation

Enter **training mode** if the input is “New User”

Creates a new profile in the database consisting of a .csv file containing user voice samples and a support vector machine (SVM) model trained from those samples. (see slide 16)

Enter **verification mode** if the input is an existing username

Checks a voice sample against a single specified profile and returns a positive or negative

Enter **identification mode** if the input is “whoiam”
Training mode

Record a user speaking a scripted phase (e.g. “bookcase”)

Acquire 5 samples for SVM’s training mode

Increase functionality with larger database

Perform silence/white space truncation from input signal

Extract features from each recording

Artificially generate larger data set for each user (see slide 15)

Use data set to train an SVM for that user (see slide 16)
Verification Mode

Take Recording

Remove whitespace/silence from recordings

Extract features

Check against SVM model associated with that user

Return result (positive or negative match)
Identification Mode

Take Recording

Remove whitespace/silence from recordings

Extract features

Checks sample against all users in the database and returns most likely user.

Closest match: user whose SVM classifier gives the largest positive classification score
Silence/Whitespace Truncation

Sets a threshold power and if the sample is below that power it is removed.

Eliminates blank space before speech so that the samples lineup in time.
Spectrogram: Before vs. After Truncation

Phil's Voice Sample
before Silence
Truncation
Feature Extraction

Create spectrogram of input signal

Divide input signal into 8 frames, 50% overlap

Apply Hamming window to reduce freq. leakage

Calculate FFT for each frame

For each frame, find frequency with highest power

A 8d feature vector is obtained by finding for each frame the frequency bin with the highest power content
Feature Extraction
Creating synthetic data points

Take feature vectors extracted from recordings

Fit normal distribution parameters (mean and variance)

\[ X_i \sim N(\text{mean}_i, \text{var}_i) \]

Generate many additional samples within those parameters
Training the Support Vector Machine (SVM)

The SVM is a binary classifier and is used for supervised machine learning.

Each user has their own SVM associated with them.

To train the SVM:

- Read in extracted features for that user. These are part of the “positive” class.
- Read in extracted features for all other users. These are part of the “negative” class.

Each feature vector is plotted as a coordinate in an n-dimensional hyperspace.

The MATLAB SVM algorithm fits a boundary between the positive and negative classes.

To verify users, a new unclassified sample is checked against the user’s SVM.
Training the SVM (MATLAB)
What does our system look like now?

Control flow of program works

Recording voice samples works

Feature extraction gives inconsistent results

Adding synthetic data points works…

But gives some extraneous results due to input variance
Why do we think our system failed?

- Even with the same person saying the same phrase, there are variances between recording instances, even for the same user.
  - E.g. saying the phrase slightly faster or slightly slower, different inflection
  - People get “impatient” during the training phase

Artificial generation of data points did not compensate instances of high variance in user samples

Normal distribution might not be the best model for generating new samples.
Example of data with high variance (left) versus low variance (right)

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>920.54</td>
<td>920.54</td>
<td>855.94</td>
<td>495.26</td>
<td>462.96</td>
</tr>
<tr>
<td>2</td>
<td>414.51</td>
<td>398.36</td>
<td>414.51</td>
<td>484.5</td>
<td>446.81</td>
</tr>
<tr>
<td>3</td>
<td>882.86</td>
<td>872.09</td>
<td>855.94</td>
<td>425.28</td>
<td>452.2</td>
</tr>
<tr>
<td>4</td>
<td>909.78</td>
<td>925.93</td>
<td>457.58</td>
<td>441.43</td>
<td>419.9</td>
</tr>
<tr>
<td>5</td>
<td>877.48</td>
<td>769.81</td>
<td>403.75</td>
<td>500.65</td>
<td>441.43</td>
</tr>
<tr>
<td>6</td>
<td>403.75</td>
<td>409.13</td>
<td>834.41</td>
<td>414.51</td>
<td>489.88</td>
</tr>
<tr>
<td>7</td>
<td>414.51</td>
<td>392.98</td>
<td>403.75</td>
<td>500.65</td>
<td>457.58</td>
</tr>
<tr>
<td>8</td>
<td>398.36</td>
<td>909.78</td>
<td>850.56</td>
<td>457.58</td>
<td>489.88</td>
</tr>
<tr>
<td>9</td>
<td>936.69</td>
<td>899.01</td>
<td>1383.5</td>
<td>446.81</td>
<td>441.43</td>
</tr>
<tr>
<td>10</td>
<td>436.05</td>
<td>419.9</td>
<td>419.9</td>
<td>506.03</td>
<td>452.2</td>
</tr>
</tbody>
</table>

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>1130.1</td>
<td>662.44</td>
<td>1086.2</td>
<td>484.21</td>
<td>475.02</td>
</tr>
<tr>
<td>12</td>
<td>359.31</td>
<td>1007</td>
<td>1159.7</td>
<td>396.21</td>
<td>451.62</td>
</tr>
<tr>
<td>13</td>
<td>489.89</td>
<td>730.99</td>
<td>411.84</td>
<td>476.82</td>
<td>468.3</td>
</tr>
<tr>
<td>14</td>
<td>723.09</td>
<td>662.09</td>
<td>269.2</td>
<td>469.24</td>
<td>444.19</td>
</tr>
<tr>
<td>15</td>
<td>326.85</td>
<td>677.66</td>
<td>619.88</td>
<td>473.84</td>
<td>476.82</td>
</tr>
<tr>
<td>16</td>
<td>809.3</td>
<td>1019</td>
<td>451.12</td>
<td>437.69</td>
<td>453.38</td>
</tr>
<tr>
<td>17</td>
<td>1019.1</td>
<td>1018.5</td>
<td>647.12</td>
<td>446.85</td>
<td>422.64</td>
</tr>
</tbody>
</table>

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>484.5</td>
<td>506.03</td>
<td>527.56</td>
<td>495.26</td>
<td>436.05</td>
</tr>
<tr>
<td>2</td>
<td>473.73</td>
<td>479.11</td>
<td>500.65</td>
<td>419.9</td>
<td>484.5</td>
</tr>
<tr>
<td>3</td>
<td>495.26</td>
<td>511.41</td>
<td>522.18</td>
<td>549.1</td>
<td>495.26</td>
</tr>
<tr>
<td>4</td>
<td>468.35</td>
<td>479.11</td>
<td>516.8</td>
<td>506.03</td>
<td>479.11</td>
</tr>
<tr>
<td>5</td>
<td>473.73</td>
<td>495.26</td>
<td>506.03</td>
<td>500.65</td>
<td>479.11</td>
</tr>
<tr>
<td>6</td>
<td>484.5</td>
<td>489.88</td>
<td>511.41</td>
<td>522.18</td>
<td>479.11</td>
</tr>
<tr>
<td>7</td>
<td>489.88</td>
<td>511.41</td>
<td>549.1</td>
<td>506.03</td>
<td>479.11</td>
</tr>
<tr>
<td>8</td>
<td>479.11</td>
<td>500.65</td>
<td>516.8</td>
<td>479.11</td>
<td>452.2</td>
</tr>
<tr>
<td>9</td>
<td>462.96</td>
<td>489.88</td>
<td>522.18</td>
<td>522.18</td>
<td>495.26</td>
</tr>
<tr>
<td>10</td>
<td>495.26</td>
<td>511.41</td>
<td>527.56</td>
<td>500.65</td>
<td>473.73</td>
</tr>
</tbody>
</table>

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>486.68</td>
<td>520.78</td>
<td>489.59</td>
<td>529.34</td>
<td>481.18</td>
</tr>
<tr>
<td>12</td>
<td>466.24</td>
<td>491.89</td>
<td>524.64</td>
<td>621.42</td>
<td>526.05</td>
</tr>
<tr>
<td>13</td>
<td>465.78</td>
<td>536.08</td>
<td>529.8</td>
<td>497.97</td>
<td>488.43</td>
</tr>
<tr>
<td>14</td>
<td>478.46</td>
<td>495.84</td>
<td>540.1</td>
<td>547.88</td>
<td>501.29</td>
</tr>
<tr>
<td>15</td>
<td>488.17</td>
<td>482.03</td>
<td>529.69</td>
<td>555.37</td>
<td>484.3</td>
</tr>
<tr>
<td>16</td>
<td>492.19</td>
<td>506.68</td>
<td>515.94</td>
<td>510.07</td>
<td>460.93</td>
</tr>
<tr>
<td>17</td>
<td>490.57</td>
<td>482.8</td>
<td>505.62</td>
<td>472.67</td>
<td>421.44</td>
</tr>
</tbody>
</table>
Where’s our hardware?

Our project became very software-oriented

Possible hardware components

  Wireless Microphone with pcb power supply

  Implementation of algorithm(s) on Raspberry Pi
## Specs

<table>
<thead>
<tr>
<th>Category</th>
<th>Proposed</th>
<th>Achieved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification accuracy</td>
<td>&gt;90% accuracy</td>
<td>&lt;50% accuracy</td>
</tr>
<tr>
<td>Database</td>
<td>SQL database</td>
<td>Unstructured data</td>
</tr>
<tr>
<td>Latency</td>
<td>&lt;1s</td>
<td>~1s</td>
</tr>
</tbody>
</table>
Moving forward...
New Specs (for post-FPR)

<table>
<thead>
<tr>
<th>Category</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification accuracy</td>
<td>Able to recognize Paul, Liam, Phil, Jason with &gt;80% accuracy</td>
</tr>
<tr>
<td>Power supply</td>
<td>2.5A @ 5V</td>
</tr>
</tbody>
</table>
Questions?