Voice Coded Lock

Electrical & Computer Engineering

Aman Thombre and Logan Greuel
Team 62

5/1/2023

Agenda

- Introduction and Objective
- Design
- Project Build
    - Hardware
    - Speech Recognition Software
- Successes and Challenges
- Conclusions

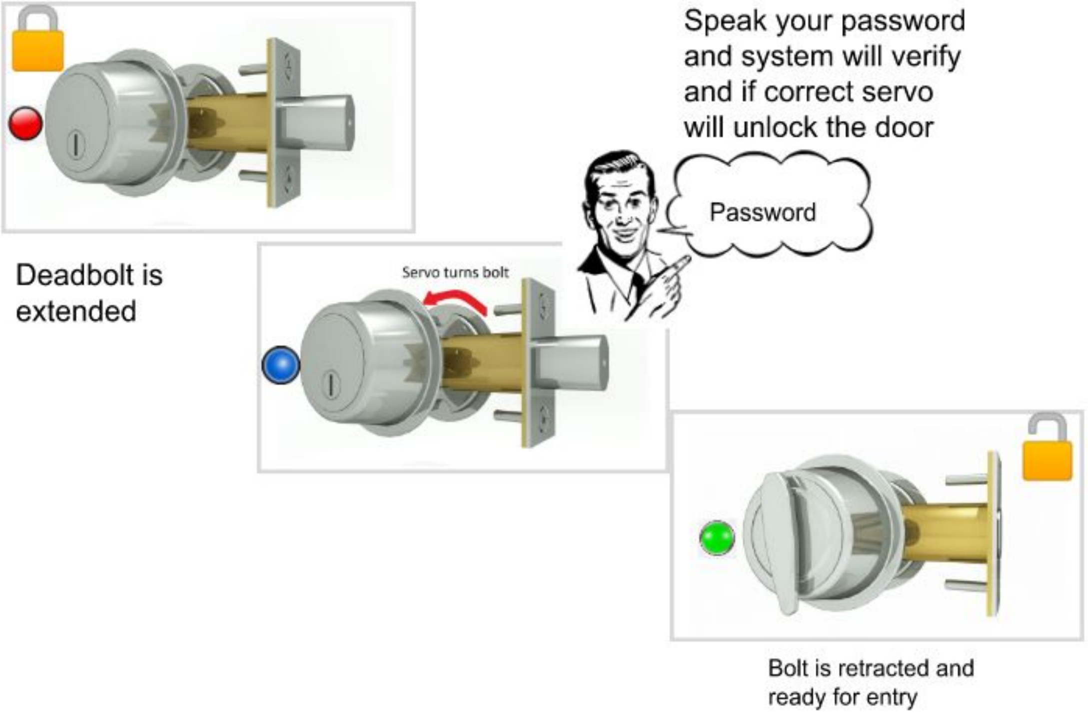
Introduction and Objective

Hands free, automatic door lock.

Use a spoken password to automatically unlock a door.

Why?

- Entering secure areas with your hands full.
- Forgetting/losing access cards.
- Intended for applications such as labs.



Design

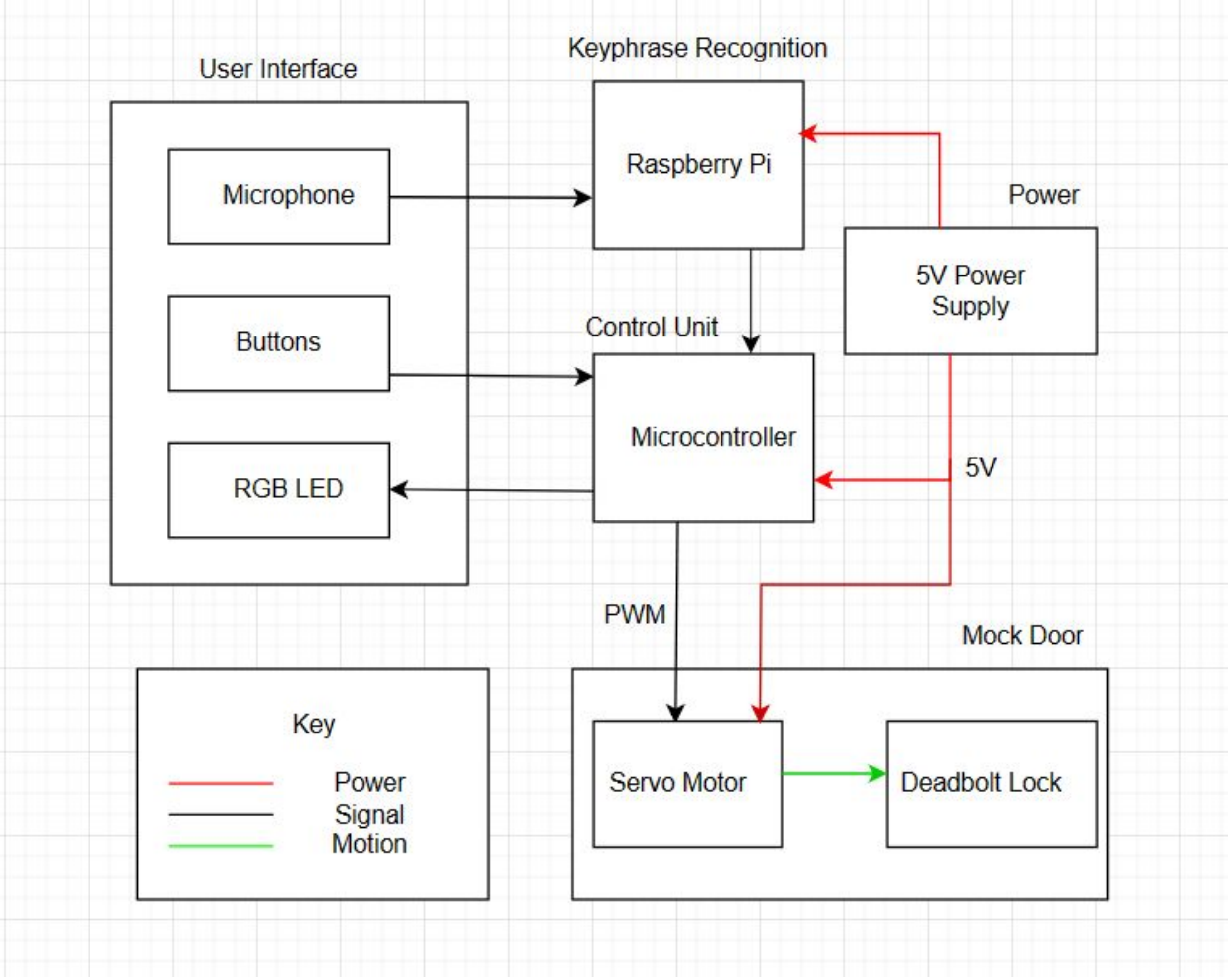
Requirements and Block Diagram

High Level Requirements

- Accurate keyphrase recognition (>80%).
- Process should take less than 8 sec.
- Automatic operation of door lock.

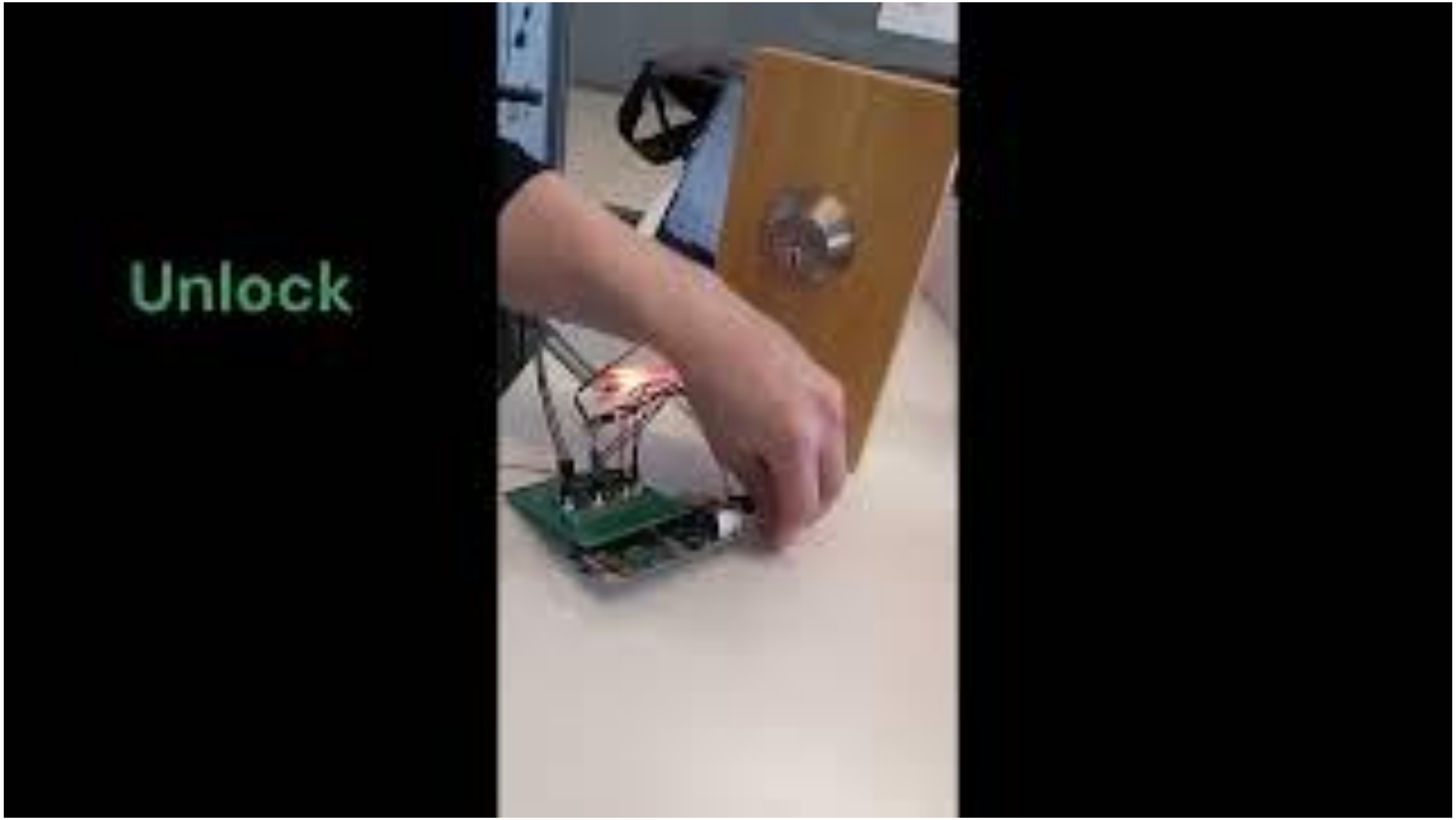
What changed in our design?

- Microphone input directly to the Raspberry Pi.



Project Build

Video



Hardware

User Interface

- Microphone directly connected to Raspberry Pi
- Buttons active low (debounce included)
- LED output, common-ground

No requirement since purchased products should function

Control Unit

- ATmega328P microcontroller
- Active low inputs, logic output for LED
- PWM output for Servo (uses counters)

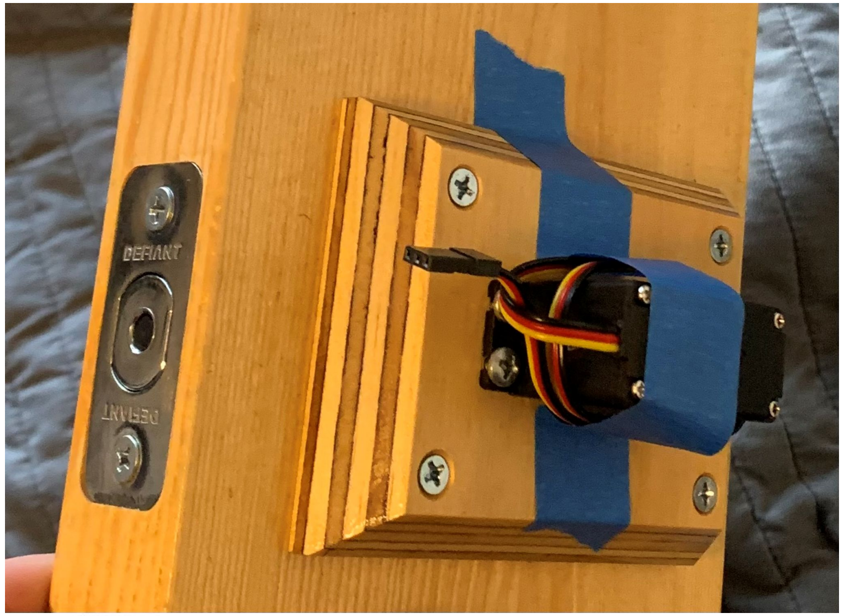
Requirement: Control inputs/outputs and send PWM signal
Verified by video, button inputs, servo and LED outputs function

Blocks cont.

Mock Door

- Built by Machine Shop
- Controlled by HS-311 Servo

Requirement: Servo must be able to lock/unlock the “door”
Verified by video



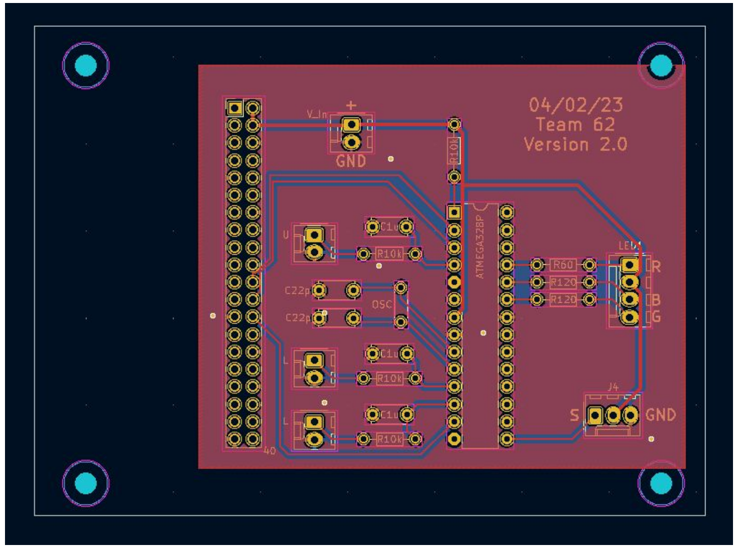
Power Supply

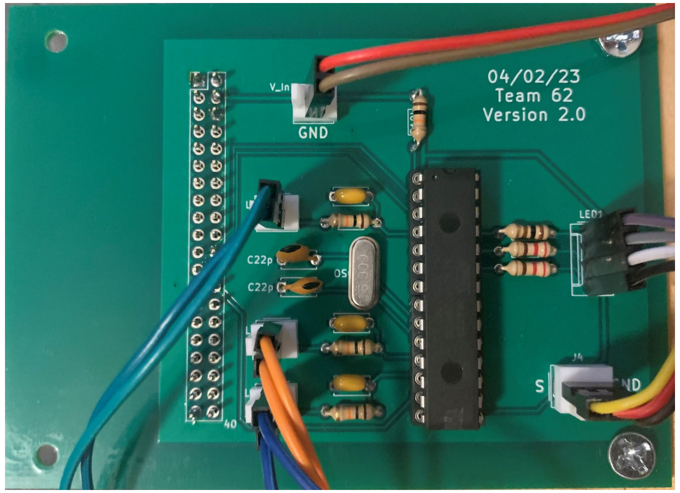
- Adapter to convert standard wall power into 5V 3A supply

No requirement since purchased product should function

PCB Components

- Microcontroller and Raspberry Pi (headers)
- Input ports for buttons with RC debounce
- Output ports for Servo/LED
- 16MHz Oscillator with Capacitors for ATMEGA





Speech Recognition Software

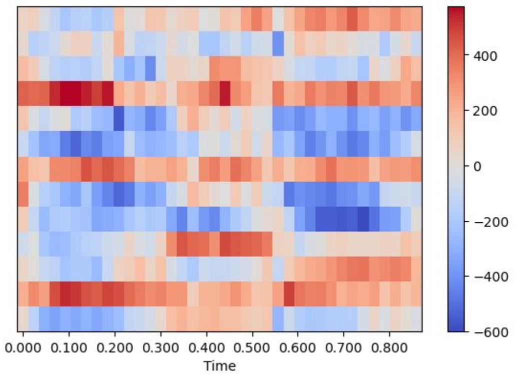
Speech Recognition Model

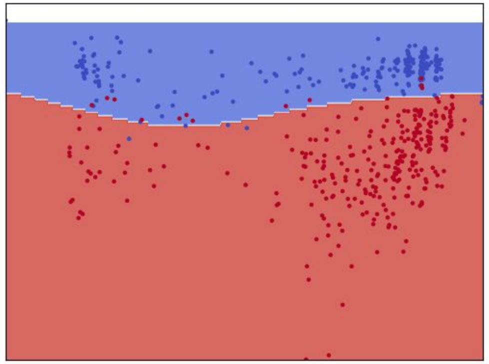
MFCC — HMM — SVM

Frequency domain information

Probability of being the password

Password or not?

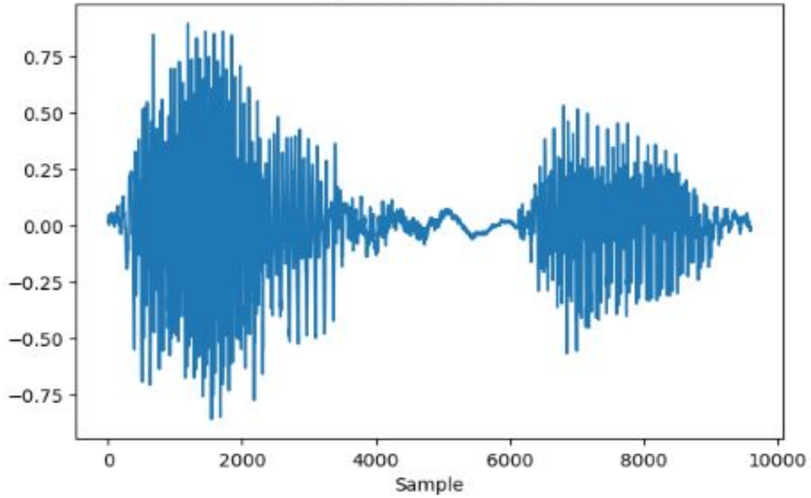


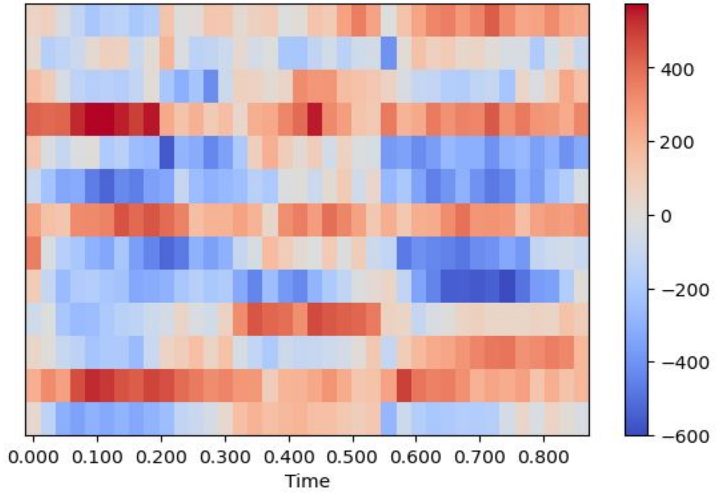


Frequency domain information: MFCCs

MFCCs allow us to capture the phonemes of words by giving frequency domain information over short periods of time.

Standard use in speech recognition, robust to noise and different speakers.





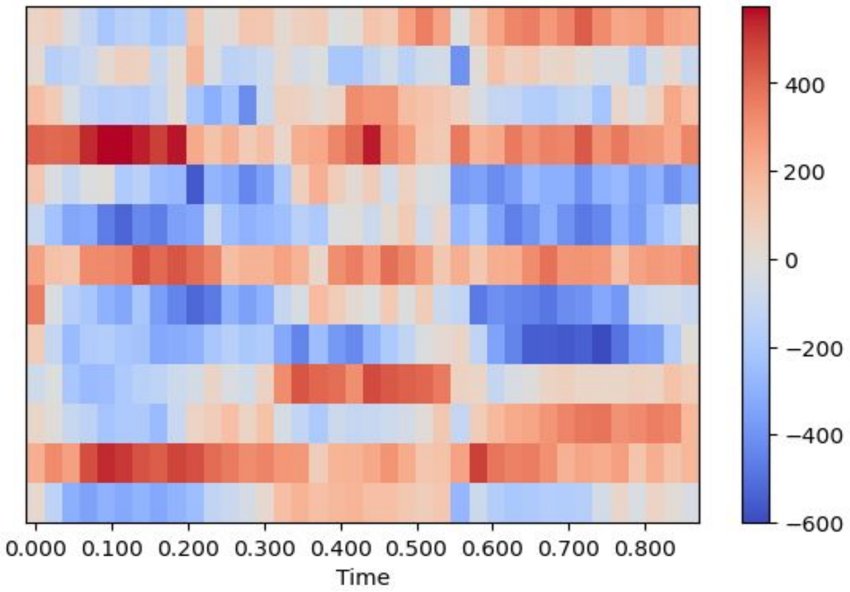
Process:

- Start recording when volume threshold reached
- Savitzky-Golay filter (SNR)
- Remove dead space (variance)
- Take MFCCs

HMMs allow us to solve for P(X|Λ).

Given knowledge of a model (Λ), we can solve the probability that our observations (X) were generated by that model.

Knowledge of the model (Λ) includes transition probabilities (A), emission probabilities (B), and initial probabilities (π).



How do we do this?

The forward algorithm:

$$\text{Define } \alpha_i(t) = P(x_1, \ldots, x_t, q_t = i|\Lambda).$$
$$\alpha_i(1) = \pi_i * B_i(x_1).$$
$$\alpha_j(t) = \textstyle\sum_i \alpha_i(t-1) * A_{ij} * B_j(x_t).$$
$$P(X|\Lambda) = \textstyle\sum_i \alpha_i(T).$$

How do we learn Λ?

Key idea: model B as a multivariate Gaussian, so Λ = {A, μ, Σ}.

Learn Λ using the Baum-Welch algorithm (MLE for HMMs).

Given training examples for each model:

- Make initial guesses for A, μ, Σ.
- Baum-Welch: forward and backward algorithms, E-step, M-step.
- Repeat with A_{new}, μ_{new}, Σ_{new} until convergence.

$$\gamma_i(t) = P(q_t = i|X, \Lambda) = \frac{\alpha_i(t) * \beta_i(t)}{\sum_i \alpha_i(t) * \beta_i(t)}.$$

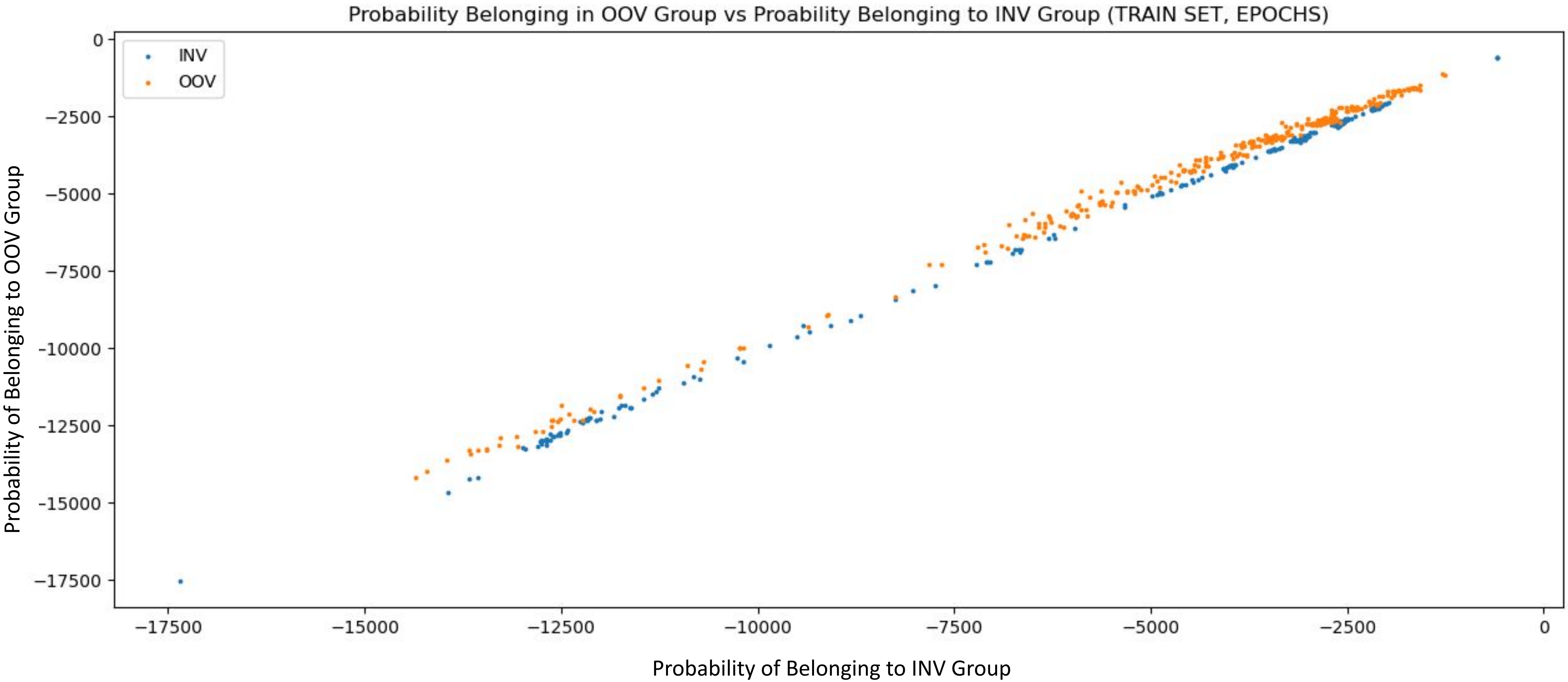
$$\xi_t(i, j) = P(q_t = i, q_{t+1} = j|X, \Lambda) = \frac{\alpha_i(t) * A_{ij} * B_j(t+1)}{\sum_{i,j} \alpha_i(t) * A_{ij} * B_j(t+1)}.$$

$$A_{ij} = \frac{\sum_t \xi_t(i,j)}{\sum_j \sum_t \xi_t(i,j)}.$$

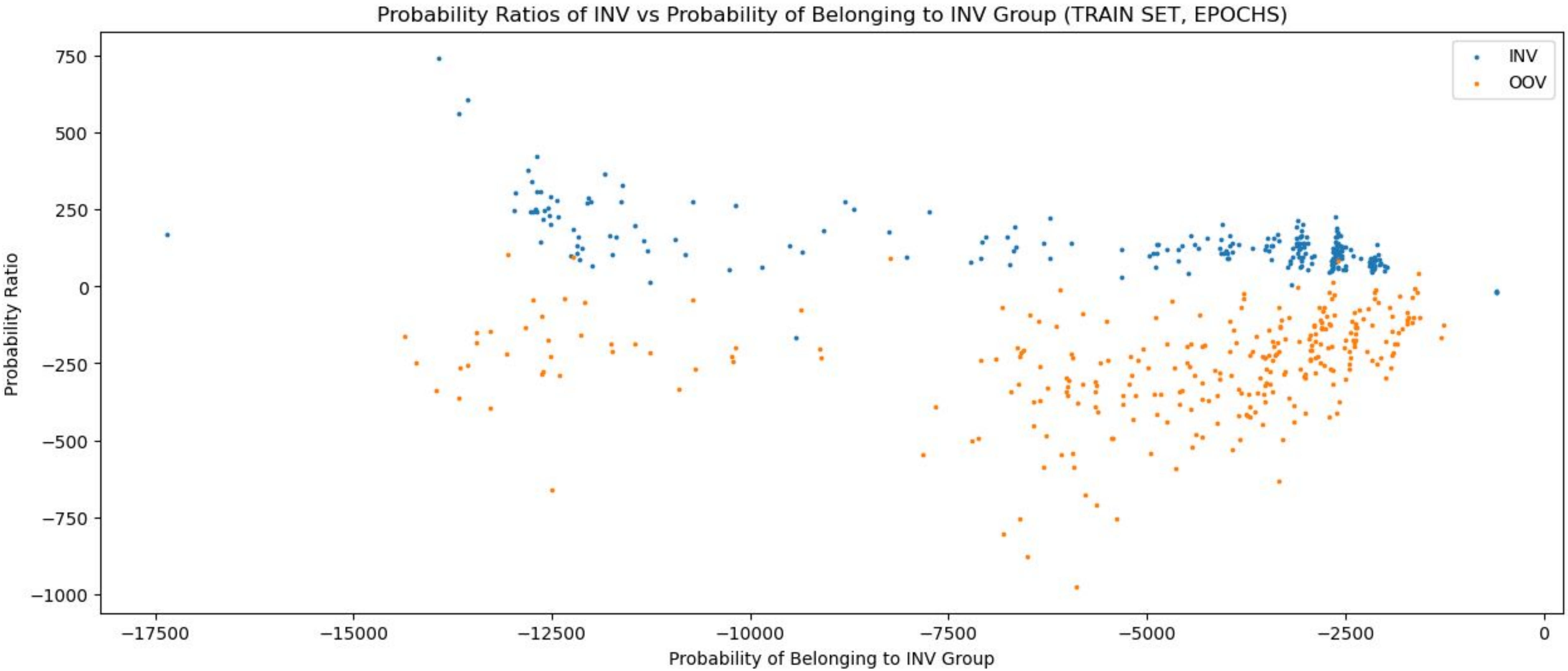
$$\mu_i = \frac{\sum_t \gamma_t(i) x_t}{\sum_t \gamma_t(i)}.$$

$$\Sigma_i = \frac{\sum_t \gamma_t(i)(x_t - \mu_i)(x_t - \mu_i)^T}{\sum_t \gamma_t(i)}.$$

HMM Graphs: Probabilities of Belonging to Each Group



HMM Graphs: Ratio of Probabilities



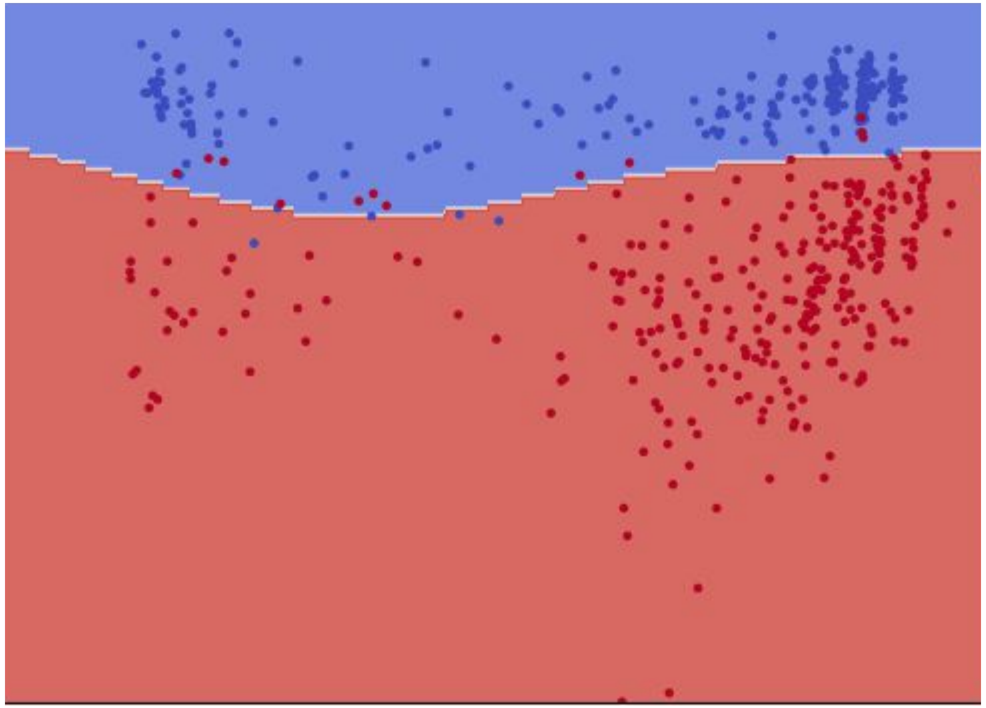
Given these probabilities, how should we classify X?

Choosing the model with the higher probability doesn't work well, especially in edge cases.

Use an SVM to place points on either side of a decision boundary.

SVM features:

- Probability of being INV.
- Probability of being OOV.
- Ratio of those two probabilities.



Training and Testing the Model: Datasets

Recorded “hamster” (INV class):

Train: 250 | Val: 50 | Test: 50

Recorded random words (OOV class):

Train: 200 | Val: 50 | Test: 50

LibriSpeech “chunks” (OOV class):

Train: 800 | Val: 50 | Test: 50

Results on the Test Dataset

		Ground Truth (spoken word)	
		INV (“Hamster”)	OOV (Random)
Classified as	INV	43	4
	OOV	7	96

- Accuracy = **0.927**
- F1 Score = **0.887**

Final Results on the Door Lock

		Ground Truth (spoken word)	
		INV (“Hamster”)	OOV (Random)
Classified as (Door action)	INV (Door unlocks)	18	3
	OOV (Door locks)	2	17

- If the door unlocks, what is the probability that the user said the right password?
  - Precision = **0.857**
- If the user says the right password, what is the probability that the door unlocks?
  - Recall = **0.9**
- If the user says a random word, what is the probability that the door locks?
  - Specificity = **0.85**
- Accuracy = **0.875**
- F1 Score = **0.878**

Successes and Challenges

Successes and Challenges

Successes

- Designed and implemented working PCB.
- Obtained Experience with microcontroller.

- Keyphrase recognition is accurate (~85%).
- Recognition works fast (~1 second).

- **Met all high level requirements.**

Challenges

- Lots of unfamiliar tasks.
- Keyphrase recognition has some sensitivity to background noise.

Conclusions

Learning Points and Future Work

Project Takeaways

- Always check the data sheets.
- Order components early and know what is needed.
- Disconnect between datasets and real-world.

Future Work

- Feedback from microcontroller to Pi for smarter listening.
- Improve aesthetics of the build design.
- More expansive and representative datasets.
- Speech recognition on a phone app - better model and audio quality.

The Grainger College of Engineering

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN