ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY OF MICHIGAN

Hands-free Typing & Text-to-Speech using Eye-Tracking

EECS 452: Digital Signal Processing Design Lab – Fall 2024
Simon Cadavid, Pengyang Wu, Haoxiang Li

Introduction

Many disabilities (e.g., facial or other muscle paralysis) affect people's ability to use electronic devices, specifically typing. Furthermore, those with facial paralysis often can't speak normally. By using eye-tracking to map eye movement onto a virtual keyboard, users will be able to type out words and sentences, hands-free, and then have it play back via text-to-speech.

Techniques

Pupil Detection: First, we use an open-source Python library called mediapipe [1] to do face landmark detection. These "facial landmarks" are pictured in Figure 2. They are produced by a pre-trained end-to-end neural network. We then select the facial landmarks corresponding to the pupil centers, i.e., two (x,y) pairs of coordinates in the camera image.
Calibration: We use a four-step calibration process that learns a perspective transformation to map the pupil centers to a cursor on the virtual keyboard (Figure 4).
Cursor Output: The perspective transform is applied to the camera feed to map the person's pupil positions to the virtual keyboard in real-time.
Text-to-Speech: Implemented with open-source Python library 'pyttsx3'.

System Setup

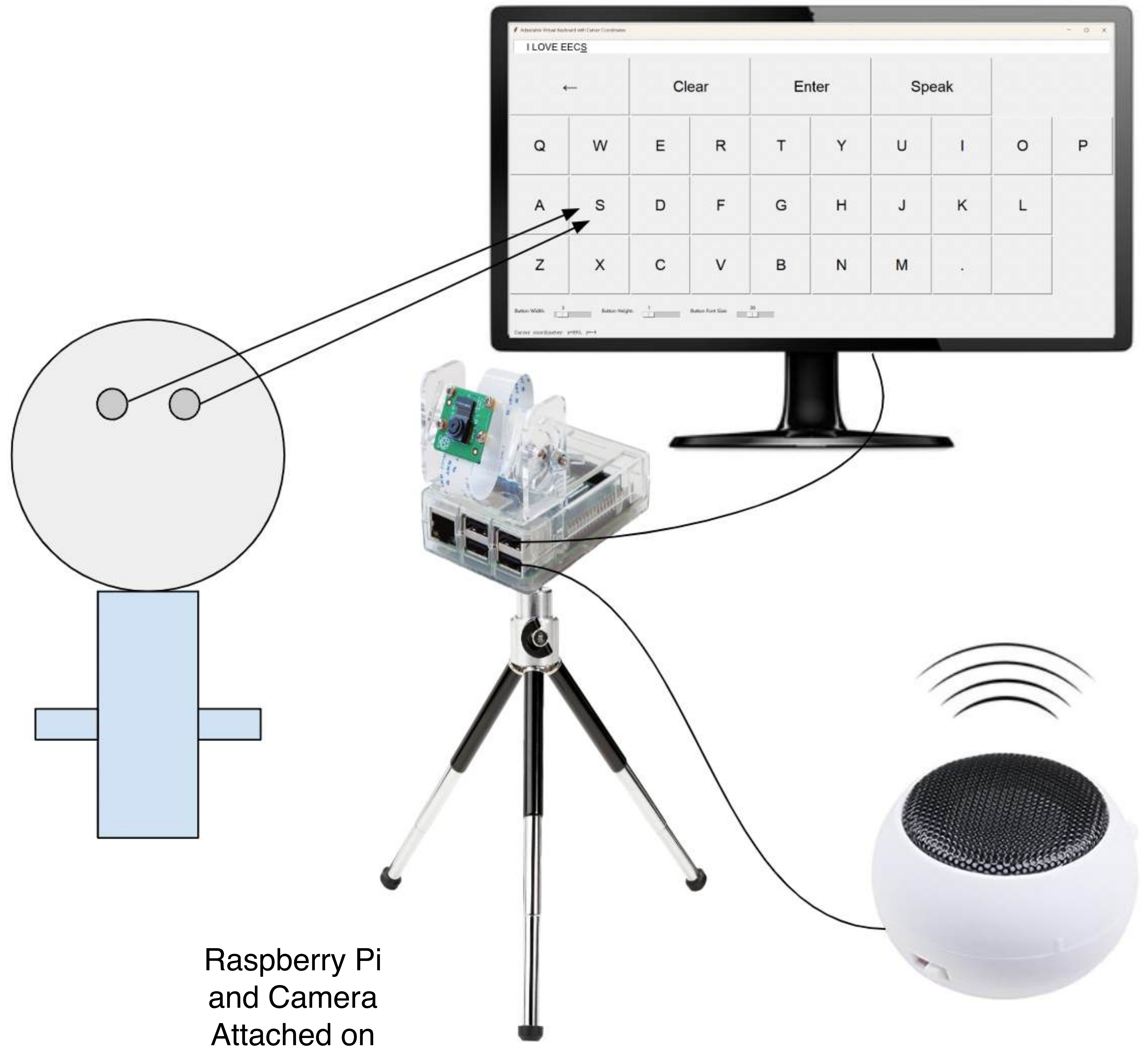


Fig 1: Final System Setup

Fig 2: Facial Landmarks from Mediapipe

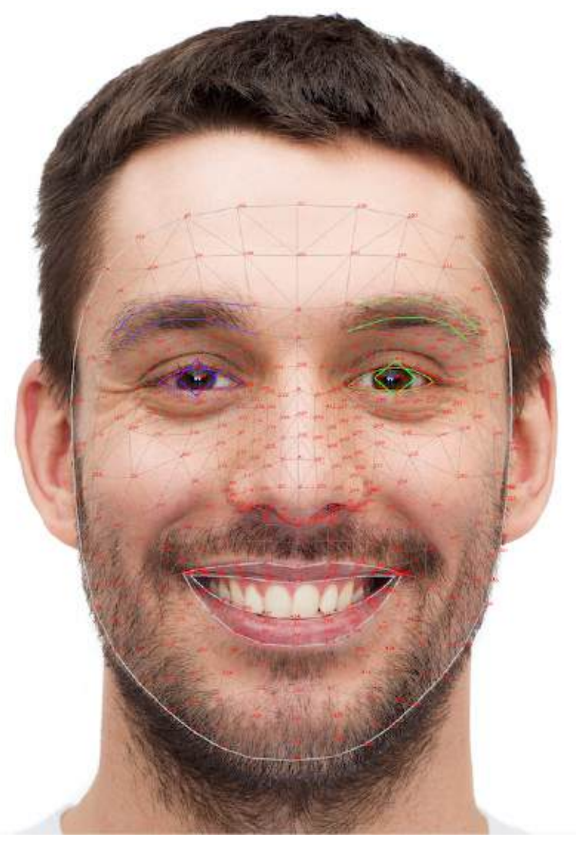
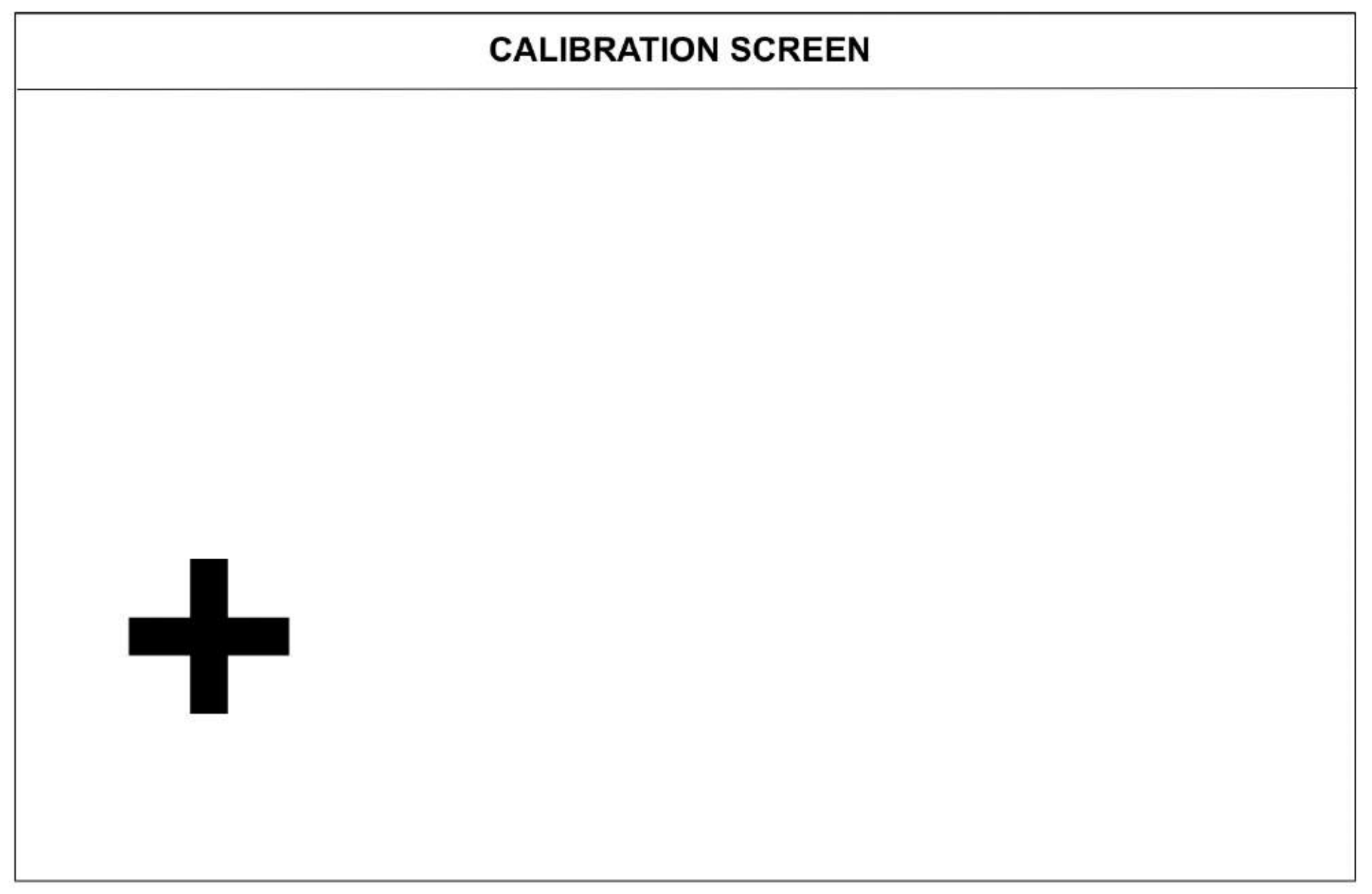


Fig 3: Pupil Positions

Fig 4: Step 1/4 of Calibration



Results

With practice, it takes about 1 minute to type and speak "HELLO WORLD" (hence, 11 CPM). The cursor noise, although we significantly reduced it through filtering, is still present, which increases the typing time.

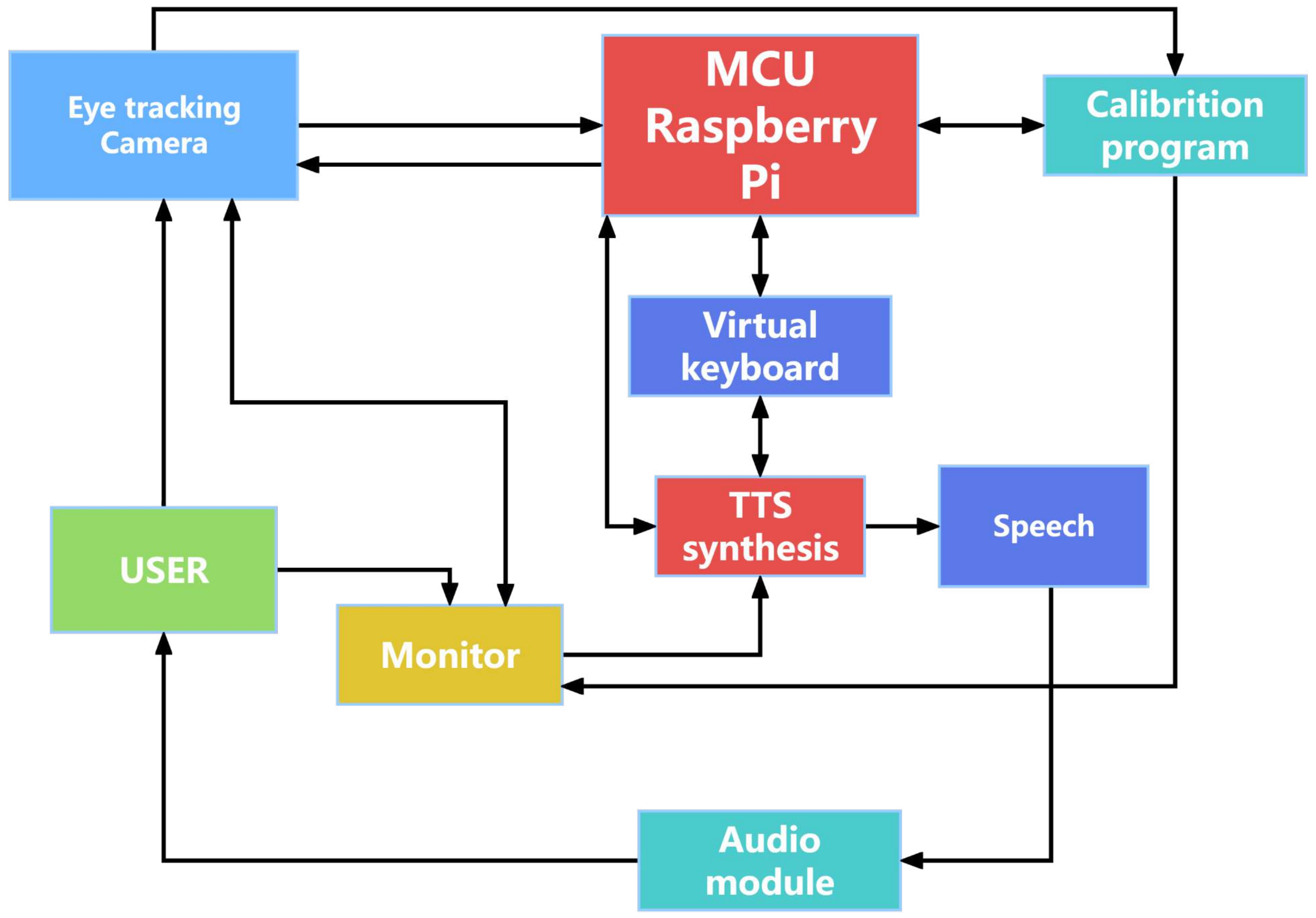
Conclusion / Future Improvements

Our system provides a relatively cheap (<$100) functional eye tracker, assuming no head or face movement, with some noise. Future improvements include using near-infrared light for better eye measurements, and a higher resolution / framerate camera for less noisy estimates. Also, more advanced (e.g., deep neural network / transformer) models for pupil -> cursor mapping.

Acknowledgements

We express our deep gratitude to Professor Jiasi Chen and GSI Cary Shu. We also thank the EECS Department for providing funding for our project.

System Architecture


Eye tracking Camera
MCU Raspberry Pi
Calibrition program
Virtual keyboard
TTS synthesis
Speech
USER
Monitor
Audio module