**Abstracts of Capstone Projects**

**December 2020**

**Viral Vision**

**Team B1:** Andrew Fish, Aditya Bhamidipati, Fizzah Shaikh, Hannah Khafif, Lucretia Field, Nathaniel Gordon

**Advisor:** Professor Bahram Shafai

**Abstract**

In the face of the COVID-19 global pandemic, the utility of high-density public spaces is reduced due to an inability to adhere to the Center for Disease Control guidelines for physical distancing. By using a low cost, open source, ceiling-mounted sensor network and motion-prediction model, this project enables the monitoring of high-traffic environments with greater respect for privacy than commercially available systems. For image analysis on the sensing device, the microcontroller utilizes the Open Source Computer Vision Library to identify subjects, then passes location data to a central processor over a Bluetooth network. The machine learning algorithms include Kalman filtering, a random forest model, and confusion matrices to train and improve the candidate motion pattern predictions for analysis by a classification model, without overfitting. A relational database stores environmental and test data for the algorithms, as well as the predictor model outputs which could trigger a light indicator to signal proximity violations.

The testing challenges presented by COVID-19 restrictions necessitated creating a motion simulation of real-life scenarios for model training. A simulated dataset modeled the typical activity of a movie theater to represent a high-volume public space. Labels in the environment indicating points of interest were used to test various movement models, such as behavioral and social, to produce potential paths. The machine learning models were trained on this simulated data for testing on real-world data.

Preserving individual privacy was an important consideration, therefore identification features were not tracked in the database. To ensure public compliance with the system, it should not be perceived as a person identification tool but rather a social modeling tool.

This project proved the ability to use low-cost hardware in conjunction with machine learning techniques to accurately predict human motion through an environment which could be harnessed for public health and security benefits. Even beyond the COVID-19 pandemic, this monitoring system could help researchers understand how human movement adapts to public health guidelines.

**Skin Conductance Activity Response for Evaluating Stress**

**Team B2:** Darin Hunt, Satwik Kamarthi, Kendall Meenan, Maxwell Nolan, Elena Silva, and Matthew Tong

**Advisor:** Professor Bahram Shafai

**Abstract**

The SCARES team has designed and built a system that can evaluate whether an individual is experiencing stress. Stress can be fairly subjective due to its wide-range of causes and differences in perception between individuals. One may find a certain class or career to be stressful while another may find the same class or job to be easy. But, there are certain measurable quantities that reflect stress, one being the Galvanic Skin Response (GSR). GSR is the variation of skin conductivity levels controlled by autonomic sympathetic activity. An example of this would be a person getting sweaty palms when they are nervous - the response is almost immediate and uncontrollable. Quantifying stress objectively would have many applications such as hospitals monitoring their patients, teachers monitoring their students, and athletes monitoring how strenuous their workout was.

The purpose of this project is to implement a machine-learning algorithm to identify whether a user is stressed and provide the processed data to specific stakeholders. The individual under test wears a GSR Sensor in an armband on the bicep with two leads that measure GSR by extending down the arm to the hand and attach to the index and ring finger via a velcro band. The GSR sensor transmits data to the NeuLog API running on the user’s computer. The raw data is forwarded to the SCARES custom API which stores it in a database and sends a message to an AWS queue, notifying the SCARES processing program that new data needs to be classified. At this point, the SCARES processing program reads the queued message and kicks off the algorithm. In the algorithm a total of 10 time- and frequency-domain features are extracted from the dataset and run through a Support Vector Machine algorithm to classify the data, and that classification is written to the database. Finally, the front-end React web app pulls the data from our API to display for the end user.

The system provides real-time stress feedback by taking data from a non-invasive sensor that transmits via a computer to a custom-built data pipeline that can show a remote party whether an individual is stressed. Ultimately, this project utilizes binary classification - it answers the question: “Is the user undergoing event-related stress right now?” The SCARES team believes that this research project can serve as the foundation for *quantifying* stress in the form of a “Stress Score” for the future.

**PLC Based Assistive Robot (PLCBAR)**

**Team B3:** Aakash Rohra, Alon Neerman, Eeswar Adluru, Jianan Dingqian, Kasra Pourrahimi, Tyler Phillips

**Advisor:** Professor Bahram Shafai

**Abstract**

The PLCBAR team has designed and built a robot to prove the concept that we can bypass the vendors software for programming logic units sold by manufacturers. The software we build will help to interface with PLCs from multiple manufacturers at the same time without the need to buy the vendor's software. This project will integrate the control capabilities of a Schneider Electric Modicon M172 Programmable Logic Controller (PLC) with a robot using our custom build software to test the full functionality of the PLC.

Using a Raspberry Pi as a network bridge between a WiFi network and the PLC, we are able to wirelessly connect to the PLC and read inputs/outputs from the registers over MODBUS TCP. The PLC is connected to an L298N motor encoder which outputs signals to 4 servo motors on the base of the robot. The speed of the robot is determined by the duty cycle of the PWM signal being outputted from the PLC. In order to actually enable the motor drivers, there is a 5V input daisy chained through the digital relays of the PLC, which control the digital relays that act as digital outputs for the encoder. When the robot needs to move forward, then both digital relays for the forward functionality are turned on as the left and right sides are controlled independently. The Raspberry Pi, along with the network bridge functionality, also reads inputs from a continuous barcode scanner and writes the barcodes to a file which is then read from the GUI. Thus, the robot can move forward and backward to a barcode scanner, read the data, and return it to the GUI without having to use any software developed from the PLC manufacturer. Powering the robot is a 26V battery connected to multiple variable LM2596 buck converters allowing us to power the PLC, Raspberry Pi, and motor encoder with the same power source. The GUI itself is a fairly minimalistic design, where there is first a login screen to connect to the PLC. Afterwards, there are options to move the robot forward, backwards, left and right. Along with these functions, there is a button to display the most recently scanned barcode, as the robot’s barcode scanner is continuously scanning.

To test the full functionalities of the PLCBAR it should be deployed in various locations such as chemical storage facilities, biohazard labs, manufacturing plants, etc. It will scan barcodes of the products placed next to each other and signal for any cross-contamination possibilities.

**AI Physical Therapy**

**Team B4:** Nikki Gharachorloo, Andrea Matellian, Allyson Vakhovskaya, Augusto Rivas Costante, Mohamed Al-Kooheji, Nathan Newbury

**Capstone Advisor:** Professor Bahram Shafai

**Abstract**

The AI Physical Therapy project is designed as an aid to optimize the physical therapy experience for patients under the supervision of their physical therapist. Patients use the band to complete their exercises, receive immediate feedback and keep track of their progress all in the convenience of their home. The exercise band communicates with an artificial intelligence application that guides users through personalized programs to aid their rehabilitation. The AI application uses a classifying algorithm which provides feedback to the users based on previous exercises and real data.

Before starting their workout, users sign up for an account, select their conditions and their strength level from dropdown options. Then, they can choose from focus areas, such as reducing tremors, building strength, and improving endurance. In addition, they can choose the duration of the workout and the body part they want to focus on. This information will generate a workout for our users which needs to be approved by a therapist. Exercise descriptions and videos are supplemental information included to help first-time users have a better idea of how to complete them.

The AI Physical Therapy band comes with a sensor that measures a user's force as they are conducting an exercise. Data such as time, force and repetitions are measurements that are being recorded in real time. As the data is being recorded it is wirelessly transmitted to our backend. A user is then able to effectively view workout results and feedback based on their movements.

Users can easily gather data with the smart resistance band as they complete any of the exercises. Then the AI PT will analyze the data and provide feedback, encouragement, and tips in a user-friendly way so users can receive clear improvement plans. By tracking metrics of force, time, and number of repetitions, AI PT will show users graphical visualizations of their rehabilitation progress. The goal is to use this application to track and manage a patient’s progress. They can then show their results to a professional physical therapist who can then evaluate the progress and adjust their treatment accordingly.

For testing purposes, 5 exercises with different targets of  hands, shoulders, hips, knees, and ankles were chosen, as also recommended by a Physical Therapy expert at Northeastern University. Data for completion of each exercise with full strength and a weaker strength were collected. Accuracy of the data collected for the number of repetition and completion time for each exercise was confirmed. In addition, how the AI app detects and groups the strength of each exercise as beginner, intermediate, and advanced was also validated.

**FindAR**

**Team B5:** Justin Chen, Takezo Johnson, Yi-En Wu, Mingsley Jiang, Michael Soria, Hsiu-Ting Yeh

**Advisor:** Professor Bahram Shafai

**Abstract**

When disasters occur, natural or otherwise, human responders are heavily relied upon in the search and rescue process. However, in the event of very large disasters or even multiple disasters striking at once, human resources can be stretched thin and effectively managing rescue efforts can quickly become a daunting task. With the aid of robotics, the burden of search and rescue tasks can be alleviated and even enhanced for humans.

We propose a system called FindAR that coordinates a team of search robots and rescue robots to be used in disaster sites to locate victims while also providing critical information to human rescue teams. The *search robot*, referred to as the Scout, can move quickly and maneuver around obstacles while relaying GPS coordinates and visual feed to quickly pinpoint casualties. The *rescue robot*, referred to as the Rover, is less agile, but can carry larger payloads and support communications between a human rescuer and casualties that it encounters. An Android application interfaces between the two by sending commands and GPS coordinates, while also providing an Augmented Reality environment to assist in rescue efforts. In this environment, beacons are overlaid onto the real world for ease of navigation to beacon locations with just the human eye. Many of these critical core functions were achieved and can be used for various applications. For example, future extensions and possibilities with single drone interacting with a robot or scenarios with multiple drones-robots are possible.

**Smart Pill Dispenser Network**

**Team J1:** Richard Conway, Damian Kulec, Gustaf Njei, William Varner, Yulu Sun

**Advisor:** John Kimani

**Abstract**

The Smart Pill Dispenser Network team focused on building a communication system between multiple smart pill dispensers. While all the existing solutions can help people with memory loss, the problem with current smart pill dispensers is that they are centralized devices. They are inherently restrictive in this nature, as one must be around the device to receive the medication. This creates a challenge if someone spends time in multiple areas of their home, needs to travel, or simply has problems with mobility. This design allows patients to have greater access, freedom, and independence by stationing dispensers in their frequently-accessed locations while dispensers communicate with one another to deliver the proper pills at the right time.

Our Smart Pill Dispenser System is user-friendly, and patients can receive their medication from the device closest to them. To use the smart pill dispensers, press the button on the nearest dispenser when notified by the device’s alarm (flashing LEDs and buzzer). The Raspberry Pi 4 controllers (inside the dispensers) will communicate with one another over the same WiFi network. The device in use will automatically update the other device of your action and it will turn off its alarm. Each dispenser has three chambers that can take three pills of different sizes. The dispensing mechanism of each chamber is made up of an interior rotating base controlled by a motor fixed underneath the exterior chamber. The pill to be dispensed fits in the opening of the rotating base, and is then rotated to the opening of the chamber where it is dispensed. The Raspberry Pi handles dispensing by sending controlled PWM signals to the DC motors. When pills are picked up and dropped, they are detected by an LED and photoresistor installed outside the opening of the chamber. Cutting off the light source to the photoresistor triggers an interrupt in the program, and the Raspberry Pi stops moving the motor.

Over the course of the semester, a functional dispenser was created to demonstrate the dispensing mechanism. A second dispenser was fully set up to represent another dispenser in the network, and was not tested to dispense like the other dispenser. Testing was conducted in the Northeastern Lab. The results indicate that the method of dispensing and device communication is valid. However, the speed of the motor and the angle to disk rotation is not completely satisfactory and could cause inaccurate dispensing. It is the belief of the team that with more tests on the dispensing mechanism, the Smart Pill Dispenser would deliver pills accurately and efficiently with the functional communication system.

**The ActivLink**

**Team J2:** Angerica Fitzmaurice, Dokken Shapero, Jason Chen, Jolie Berner, Luc Weidell, Tanvi Daga

**Advisor:** Professor John Kimani

**Abstract**

The purpose of this project is to implement a real time video conferencing system for trainers to hold exercise classes remotely. Due to COVID-19, many gyms and public areas have closed. Even if classes were not cancelled, they were being continued on platforms ill-designed for the class format. The ActivLink provides a new platform for these activities to be continued in a manner closer to that of those classes conducted in person. The platform is based on a TV soundbar like device which will connect users to trainers through a tailored backend video conferencing system with features such has “raise hand”, synchronized music, and human detection to facilitate an easier workout environment. The hardware for this platform is comprised of a single board computer and auxiliary microcontrollers for audio processing. The platform also makes use of a beamforming microphone array and CSI interface camera system.

In order to implement the video conferencing system, Twilio, a cloud communications platform built on WebRTC, was used. An Android application for trainees and a web application for the trainer, was developed. In addition to basic video conferencing functionality, such as video and audio streams, the Twilio DataTrack API was used to send messages between trainer and trainees. In the start of the class, the trainer and trainees join a Twilio group room, hosted in the cloud. The process of connecting to a room also involves access token generation, which was automated for end users. Additionally, while a trainer can see all trainees, the trainees can only see the trainer. This was accomplished using Twilio’s Track Subscription API, ensuring trainees only subscribed to the trainer’s audio, video, and data tracks. The Android app allows trainees to see all information about the trainer’s workout routine (exercises and times) as well as buttons to mute, stop video, and raise hand to ask questions. The web app for trainers, built on a bootstrap-like CSS library, also displays workout routine information as well as an option to programming a workout from the webpage itself. Once connected to the room, the trainer starts the workout, which sends a message to all trainees (via data tracks) and automatically starts workout timers, and a locally downloaded playlist on both the Android and web app. When a trainee has a question, clicking the raise hand button sends a message to the Trainer app, and displays and “answer question” button under the respective trainee’s video. When either trainee or trainer speak, music volumes are lowered, and each participant is unmuted automatically. Human detection was implemented in the web app using PoseNet and ml5.js to detect if a participant was present. If the web app detects that a trainee has stepped away, the trainee app is notified. On both the android app and web app, if the user steps away, the volume of the music will go to zero.

For testing and validation, each of the features was tested in a video call between 2 trainees and one trainer. While all features were successfully demonstrated, it is the belief of the ActivLink team that a more comprehensive evaluation of system performance could be done with more users.

**MIND GAMING**

**Team J3**: Sultan Alzaabi, Matthew Heim, Christopher Zarba, Michael Zeleznik, Peter Zeqo

**Advisor**: Professor John Kimani

**Abstract**

The Pokemon BCI team has designed and built a system for brain-controlled inputs to a video game that, using sufficiently accurate EEG data, can allow a user to control the inputs for the game Pokemon Red with their mind and eyes. This system is trained using sample data from the user to recognize and classify their brain waves while they look at a control input displayed on the computer screen. This technology is useful for people with physical disabilities, like ALS, who cannot move muscles voluntarily except for the eyes and might struggle with a different option like a breath-controlled device.

For this system, the user wears an EEG headset and looks at a monitor displaying the game. The game is displayed in the center of the monitor and is surrounded by visual cues representing each of the possible inputs to the game. Each visual cue is a checkerboard pattern flashing at a different frequency. The visual cortex emulates the frequency of the flashing pattern it observes at a given moment. This phenomenon is called visually evoked potential (VEP) that allows the system to classify visual cues as game inputs based on the frequency of the VEP signal.

This system has been tested for up to 8 targets, with each stimulus cycle at one second. This means that the user is able to look at a target for one second in order to choose their input. It may have somewhat increased accuracy with less targets, but currently it is configured to 8 since that is the designed number of possible inputs to the Pokemon game at any given time. It is designed to be flexible with regards to who the user is, but calibration is needed for a new user.

The main problem encountered in this project was faulty hardware. The Emotiv EPOC+ bought for this experiment turned out to not be accurate enough to record EEG data because there was not a high enough power at each target frequency. The ML models we used were not able to make accurate predictions because there was no VEP signal with this EEG. While the group is confident in the integrity of the software model that has been designed, the lack of usable EEG data means that if this project were planned for the future, the next step would be to acquire a better headset and test the model with real EEG data. The model has shown to be able to classify simulated data, created from sine waves with multiple sources of noise to simulate actual EEG data. We were able to play the game using the keyboard, generating noisy VEP signals for one of eight target frequencies.

**ASL Assistant (A2)**

**Team J4:** Jingyue Chen, Maxim Kim, Dominic Ridley, Derek Feng, Jason Suwandi

**Advisor:**John Kimani

**Abstract**

ASL Assistant (American Sign Language Assistant) is a modern and easily accessible solution to facilitate communication with people with hearing disabilities. People with hearing disabilities are prone to a variety of psychological issues due to difficulties communicating with a vast majority of people, which we aim to resolve through our developed solution. ASL Assistant, or A2for short, allows for translation of sign language into text employing machine learning algorithms and data science concepts. ASL Assistant’s design concept consists of an easy-to-use website where users can click a single start-session button, prompting the browser to accept input from the user’s webcam and record hand gesture data. The data is then sent to our pretrained model, which translates American Sign Language fingerspelling input to textual output. The model sends the result to the front-end where the user can see the output text on the right side of the camera input within the interface.

A2 consists of four software blocks – user interface, classification model implementation, pre-processing of our datasets, and feature extraction. The user interface (UI) is designed to be intuitive for the average user, consisting of just a few elements, the first of which is a window which shows live capture data from the camera. Additionally, a button is present below it which allows the user to start and stop the live translation session, and lastly a transcription column on the side that shows the translation of the current input.

Our model a multi-layer perceptron classification model that is fed an input layer of 21 hand landmarks that have been returned from the Mediapipe Handpose library. These landmarks are composed of x, y, and z values from static images, which correspond to the bounding boxes around the hand found in the image and the palm point found in the hand, respectively. The final neural network model is trained using 21,000 self-collected images sampled from recorded videos. The images fall into 29 classes; 26 are ASL fingerspelling letters, while the remaining 3 are “delete”, “space”, and “nothing”.

Originally, the model we constructed from the ground up was a convolutional neural network, which we planned to use since it predicted quite accurately on the validation sets during the training process. However, the accuracy of the model when evaluating previously-unseen data (the expected use case of the application) was unsatisfactory, as the prediction accuracy was only approximately 38 percent. To resolve this, we tried different approaches for building the model including transfer learning and using the Handpose library, the latter of which proved to increase accuracy the most, with a model validation accuracy of around 85% on our testing set.

We determined that the current state of the application serves as a proof-of-concept for expansion upon. Next steps for improvement include increasing accuracy of the prediction model, as well as dedicated hosting for the application in order to serve distributed users.

**phuudy**

**Team J5:** David Kramer, Stephen Boyd, Chris Natcharian, Matthew Shako

**Advisor:** Professor John Kimani

# **Abstract**

The phuudy (pronounced “foodie”) team designed and constructed a device and mobile app for monitoring the status of perishables within a refrigerator based on the presence of gasses associated with food spoilage. The device aims to cut down on food waste by preventing people from throwing out food that is safe to eat, and improve a user’s health by notifying them about possible food spoilage.

Phuudy detects ammonia (NH3) using a metal-oxide semiconductor thin-film sensor (MQ-135) and a temperature and humidity sensor (DHT-22). The presence of ammonia is associated with microorganisms which cause food spoilage to occur. The sensor data is filtered and amplified before being converted to digital signal and stored in a Raspberry Pi 4, where it is then evaluated against our spoilage model using machine learning. Once the spoilage status of the food is determined, the data is transferred from a connected Arduino Nano 33 BLE over Bluetooth to our mobile app for view by the user. The phuudy device stationed within the refrigerator is controlled by the Arduino to minimize power usage, effectively maintaining a ‘sleep’ mode for the Raspberry Pi.

Machine learning code, written in Python, is used for spoilage identification - determined by the training model. It works by running the training data through multiple regression models (k-NN, SVM, Decision Tree Classifier, and Random Forest Classifier) to find the model that can best predict the outcomes. Then, the most accurate model is used to make predictions for the new data collected from the sensors. When the model makes a prediction, it uses the accuracy score as the confidence value. The confidence value represents how accurate the prediction is and then is rounded up. Prior to the data being used in the model and identified, it needed to be cleaned and filtered. This includes making sure there is consistent formatting throughout and that the breaks that separate the data are recognized by the Pandas library.

The Arduino is in charge of managing the power consumption of the device. The Arduino first lets the Raspberry Pi run for ten minutes to gather usable data and produce a confidence value representing the confidence that food is spoiled. When the Arduino receives serial input of the confidence value from the Pi, it uses it as the value of the bluetooth characteristic it is advertising. After ten minutes, the Arduino shuts the Pi down completely for the next fifty minutes and only uses its power to advertise the bluetooth signal our app is searching for. The Arduino then turns the Pi on again to record new data and calculate a new confidence value.

Phuudy comes with a companion mobile application developed for iOS. This app connects to the device over Bluetooth and displays the percent chance of spoilage in the fridge to the user. It can notify the user from the background when this chance is sufficiently high (above 70%). The app also incorporates a grocery list, as requested from our survey, which allows users to keep track of what food they have in their fridge and when it expires. Also, if notifications are enabled, the Phuudy app can alert the user of the closest expiration date on their list each day.

**Acoustic and Vibration-Based UAV Preflight Diagnostic System**

**Team C1:** Nikhil Bhat, Jeremy Venne, Shane Treadway, Tommy Ma, Julian Lechner

**Advisor:** Professor Charles Dimarzio

**Abstract**

Preflight inspection is required by the Federal Aviation Administration before every Unmanned Aerial Vehicle (UAV) flight. An automated system for UAV inspection would serve to make the entire flight process unmanned, which would save companies time, training resources, and money. For this capstone project, we developed a low-cost hardware and software solution that could solve the issue of manual preflight inspections. This project is in collaboration with GreenSight, a Boston-based drone startup.

For our capstone, we chose to focus solely on identifying propeller defects preventing the drone from having a safe flight. This served to be an extremely useful building block for GreenSight, and allowed us to narrow the scope of our project. We developed a Raspberry Pi-based audio data collection system, and a vision-based machine-learning algorithm which automatically classifies spectrograms of propeller audio to detect whether the propellor is damaged.

GreenSight’s drones return to a storage station between missions, providing a site for integrating this product. Rather than having an external excitation source, we are able to leverage the drone’s on-board propellers to produce an acoustic sample. An analog microphone connected to a Raspberry Pi records the acoustic information throughout the testing period. The ML system then classifies the audio file as representing the audio from a propeller with no, moderate, or severe damage.

Physically, the two propellers are different in geometry. While an undamaged propeller is symmetrical about its center, resulting in smoother, more laminar airflow when it spins, a damaged propeller is asymmetrical, resulting in more turbulent airflow. These differences are noticeable both when listening to the audio signal, and when looking at the FFTs.

There is already a tremendous amount of background research in computer vision techniques for object classification. Rather than developing a completely novel approach for sound classification which takes in audio input, we decided to leverage an existing computer vision framework for image classification and used VGG16, a state-of-the art image model for feature extraction. To utilize an image classifier, we need to transform the audio signal into an image. For this we utilize the spectrogram, which is a visual representation of an audio signal that contains both the frequency and time domain information.

Our machine learning model performance was impressive. Evaluating the model using the validation dataset, the model has nearly 100% accuracy in detecting a good propeller, a minor damage propeller, or a badly damaged propeller. These results can be attributed to the distinctive, well-defined features present in the spectrogram. The linear model can easily converge and obtain a near perfect result. When testing the model in real-world scenarios, it still retained 97% overall accuracy, and has 100% recall rate for both damaged cases. This means all damaged propellers are successfully identified, which is the most important criteria for flight safety.

The primary impact of this project lies in eliminating the need for these manual drone inspections; however, the implications of this technology are greater than that. Such a system can be leveraged to produce automated inspection of any autonomous system, allowing for great reusability in industries like manufacturing, defense, or transportation.

**C2 Smart Glasses**

**Team C2:** Thomas Kaunzinger, Lauren Javan, Mohammed Laota, Juliann Liang, Andrew Nedea

**Advisor:** Charles DiMarzio

**Abstract**

The C2 Smart Glasses were created in order to provide the user with a real-time transcription of the conversation they are having. The goal of this device is to make sure that those that are hard of hearing can be an active participant in any conversation. The flow of conversation can be difficult to follow due to the speed of conversation or if the other person doesn’t speak loudly or clearly enough. The smart glasses hope to ease some of these difficulties. In order to do so, the smart glasses device employs Bluetooth, optics, mechanical / electrical design, and Google’s transcription API.

The main hardware components consist of a Raspberry Pi Zero, a rechargeable battery module, an organic light-emitting diode (OLED) display, and an omnidirectional microphone. The Raspberry Pi handles the basic interaction between the hardware and software components. It is loaded with software that ties together the OLED, Bluetooth, and microphone functionality. Upon boot, the raspberry pi will launch its software, and wait for a connection to an Android device to communicate with it and provide / receive audio data / transcriptions.

The Pi uses I2S to read in audio samples from the MEMS microphone embedded within the device’s enclosure. These samples are relayed to the Android mobile application via the Bluetooth connection. The app uses WiFi or cellular connection to send the samples to Google transcription services. Once the transcription is complete, the app will send back the information to Pi using the Bluetooth connection. The Pi processes the packets and sends the data to the OLED screen, which will create the visual captions for the user by way of optics. This approach can easily be expanded to leverage a language translation API, which would further enhance the device.

The subtitles are projected onto a semi-transparent acrylic mirror screen using basic optics. The OLED has the image of the subtitles and it is reflected onto a mirror at a 90° angle within the device. The image is then reflected onto another, now transparent, mirror, again at a 90° angle. This allows the image to be refracted onto the screen for the user. Between the mirrors, there is a small lens, which is used for both making the image from the small OLED screen much larger, but also projecting the image further in space from the user’s eye to prevent blurriness and eye strain.

**Persistence of Vision Display**

**Team C3:** Becker Ewing, Emma Brodigan, Monika Pesa, Sam Martel, Taha Vasowalla, Rukmini Dasadhikari

**Advisor:** Professor Charles Dimarzio

**Abstract**

The POV team created a persistence of vision display. Persistence of vision is a phenomenon caused by latency in human visual perception. When visual stimuli projected onto the retina changes fast enough, the signal is perceived as one integrated image. This phenomenon is able to take place thanks to the effects of afterimages and temporal integration and has been a subject of research and application in various technologies. The POV team’s goal was to build a device that generates a persistence of vision illusion with quickly moving LEDs. Besides being a visually exciting, challenging design puzzle, this type of device has potential application in physiological and neurological research in the areas of visual perception and processing. The POV team created a device which uses a rapidly rotating array of LEDs to generate an illusion of a continuous circular image. Software was developed to allow for users to display custom images and control the speed of rotation.

The POV display consists of a small motor, housed by a custom 3D-printed mount, that rotates two lightweight 3D-printed plastic arms covered in LED strips. The LED strips are powered and driven via wires through a slip ring coupled to the motor shaft. An esp32 microcontroller runs a program that transforms and transmits image data to multiple smart DotStar LED strips. Taking a matrix of pixel values, this program dynamically maps these values to the appropriate physical LEDs to create the illusion of a still image. A Hall effect sensor connected to the MCU, together with a magnet mounted to one of the arms, is used to track rotation speed and prevent the image from drifting. This allows the MCU program to send RGB data to the LED strips at carefully timed intervals such that the POV illusion is maintained. A Raspberry Pi 4 hosts a RESTful API that can collect images from the user, process them into an appropriate format for display, and control the rotational speed of the display. Additionally, the user can interact with this REST API via a React web application for ease of use.

For testing and validation, multiple images were processed and displayed on the device in a dark room. The POV device displayed the exact images that were uploaded using the persistence of vision effect. In the future, it’s the belief of the POV team that the software could be expanded to support displaying video and that a custom-designed PCB could vastly simplify the hardware weight and reliability.

**Tertofeel: Feel the Music**

**Team C4:** Aidan Bradley, Ethan Davis, Jason Ebbs, Harrison Gieraltowski, Lucas Ingalls

**Advisor:** Professor Charles DiMarzio

**Abstract**

The Tetrofeel team has designed and created an educational tool and engaging toy for young members of the blind and visually impaired communities. Though tailored specifically for these communities, it remains an exciting and educational tool for other children and individuals who have reason to engage with it, allowing for the honing of one’s tactile recognition skills and deductive reasoning. This product has implications and uses purely for and within institutions geared towards the care for and education of the blind and visually impaired, but based on the specifications we worked out with Perkins School for the Blind, we believe we have created an immensely useful tool in the betterment of an underserved community.

To begin play, a child or educator must turn on the unit with a bright and obvious rocker switch in the upper-lefthand corner of the unit, at which time the unit will give voice instructions to the child and/or educator on the function and application of the game. After these instructions are complete, the child or educator may shift between modes by pressing the mode button in the upper-righthand corner of the unit. All buttons were chosen and implemented with the blind and visually impaired in mind, and may be identified by a student who is blind or visually impaired with or without significant aid. The object of the game in any mode is to place all the pieces, or tetrominoes, included with the board in their respective alcoves on the game board. Only one of the game modes require these pieces be placed in a specific order. In this mode, the order in which the pieces are expected to be placed is displayed by virtue of a modulating block system towards the top of the board, one with 16 individually actuatable blocks that move up and down and sit on springs. These blocks, or “keys” as we call them, together form a “keyboard” that, when certain blocks are down, creates the shape of the piece the child must place on the board next in the order. The blocks are designed so that a child who is blind or visually impaired may easily identify the shape being displayed. The keys are kept down by electromagnets, which are activated or deactivated with relays depending on what block should be displayed. The tetrominoes have small, rare-earth magnets which interact with Hall sensors embedded in the board to let a Raspberry Pi board know that a piece has been placed, and the next piece should be displayed. Each tetromino placed correctly will allow the Raspberry Pi to add an instrument to a song that is playing - when all tetrominoes are placed correctly, the full song with all it’s instruments will play.

Given the present pandemic, we felt it unwise to spend time in close quarters with the children at Perkins School, and so instead all testing was done by Harrison Gieraltowski, who completed the bulk of the final building and implementation processes. The product works exactly as expected, and if Harrison is any judge, is significantly engaging and enjoyable. We believe that our product, were it to go into mass-production, would be an exquisite tool in the education and enrichment of the blind and visually impaired communities.

**JadePod**

**Team C5:** Konstantin Rezchikov, Chirayu Jain, Nick Pechie, Zach Stanziano, Brad Beckert

**Advisor:** Professor Chuck Dimarzio

**Abstract**

The JadePod team designed and built a desktop plant growing system to validate and explore emerging technologies in Controlled Environment Agriculture (CEA). CEA has emerged as an increasingly viable alternative to conventional outdoor farms. We knew that the limitations on Capstone and in-person meetings due to COVID-19 could make creating a large system challenging. With this in mind, we decided to create a smaller desktop prototype that would only grow a single plant, but that would also contain many of the same technologies as their larger industrial counterparts. The end result was the Jadepod, a triangular prism containing an array of sensors capable of streaming data to our onboard Raspberry Pi. The Raspberry Pi serves as our data hub, passing information to our AWS database, which is fully accessible by our companion mobile app; letting users access their growth data and improve their growing environment.

The JadePod was designed to be scalable, easy to manufacture and easy to use for researchers. The frame of the entire system was constructed primarily using off-the-shelf aluminum extrusions. All joints and hinges were then designed to be easily 3D printed without additional support material. The walls of the JadePod are rectangular laser cut acrylic panels making the entire system easy to build. The sensors chosen for the system have existing libraries to foster seamless integration with the Raspberry Pi. The JadePod’s water system is kept far from our electronics network to guard against water damage. The electronics are designed for a closed feedback sensor loop to tailor the plant’s daily light and water intake.

The JadePod runs a cron job on startup to connect to our backend and automatically collects sensor data on set time intervals. Then, the Jadepod talks to our AWS hosted lambda which updates our DynamoDB table with the current sensor data, which is displayed in our mobile application. We decided to include historical data for some sensors so a user could track how different factors affected their plant. We display the last uploaded image of our plant to the app so users can view their plants in nearly real time. Lastly, we added functionality for working with multiple JadePods, giving us the option to expand the JadePod network in the future.

One of our main takeaways was the importance of lessening the computational load on physical hardware when cloud computation was plentiful. As we moved our code onto our AWS backend, we saw increased load speeds within our application, improving the user experience.

We tested that our Pi and sensors were monitoring the plant and maintaining optimal growth conditions over the last few weeks of the semester. Our snake plant survived, and continued to grow while inside the pod, if not perhaps slightly blinded by our grow lights. We want to verify this behavior with more species of plants and over longer stretches of time, making meaningful use of the data we’ve collected. The ability to control the JadePod from the mobile application was a goal of ours and would be possible with future app updates. We wanted to dive into frequency spectrum analysis using our IR camera to inform us of water distribution patterns, IR radiation patterns, heat maps, nutrient maps, and chlorophyll distribution, but we ended up having to focus on the core system features before we could get there. Future research with this device could easily incorporate that type of analysis thanks to the onboard hardware that’s already included.

**Partial Fault Detection Algorithm for Multi-Agent Systems**

**Team M1:** Jason Allen, Tiffany Chan, Sarah Coffen, Suzanne Cuozzo, Yamina Katariya

**Advisor:** Professor Masoud Salehi

**Github Repository:** <https://github.com/erronknight/M1-ECE-Capstone>

**Abstract**

Partially failed robots are a challenge in swarm robotics, as these robots may transmit faulty data that compromises the integrity and effectiveness of the swarm as a whole. We defined a partially failed robot as an individual robot that contributes to the collaborative behaviors of a swarm which experiences damage to a particular area (sensor, motor, etc), leaving it responsive but operating in an unintended way.

Our project discusses an algorithm to detect partially failed robots, scale a swarm accordingly, and repair the swarm as a whole so that it can continue its mission. By “repair,” we mean to reconfigure the swarm to operate without the partially failed robot. In the case of monitoring a forest fire, implementation of this algorithm will ensure quick, accurate data for mapping fires without being compromised by partially failed robots. The partial fault detection algorithm iteratively checks each robot in the swarm for faults in its Laser Scanner, IMU, and Odometer. We use an expected value +/- tolerance model to determine an acceptable range for this sensor data; if the robot is found to be transmitting values outside of this range, the robot will be classified as partially failed and will be removed from the swarm.

Due to the presence of multiple robots in our swarm, each robot is indexed within the navigation stack such that there are unique *move\_base* related nodes. Additionally, the individual maps for all robots need to be merged. Navigation goals can be provided in Rviz using the 2D navigation goals feature, and an additional Frontier Exploration package was also considered for mapping unknown territory. A navigation goal is sent to each individual robot through the namespaced *move\_base* and each robot can navigate and map throughout the environment. Each robot first builds their own map using their available sensors and localizes within it using SLAM techniques. Then these are merged and used as the frame of reference for goal-based motion.

When moving as a swarm, a single position is used to generate *move\_base* goals that are sent to each robot’s navigation stack. Path planners and costmaps are used by the navigation stack to create a plan that can be dynamically altered as the robots learn more about their environment or if they encounter obstacles. If a robot is determined to be in a failed state, they attempt to return to its starting position, if the robot can still move.

For testing our partial fault detection algorithm, we developed error injector scripts for the Laser Scanner, IMU, and Odometer. These scripts are capable of adding noise to the sensor data, effectively simulating sensor faults.

For our project, we use Gazebo for simulation and the Robot Operating System (ROS) with python scripts for programming and controlling our robots. The robots simulated are the Turtlebot3 Burger models. These were run inside an Ubuntu 16.04 environment.

**SANDBot**

***S*emi-*A*utonomous *N*ature *D*e-littering ro*B*ot**

**Team M2:** Joseph Straceski, Jason Fitch, Noah Hamlen, Liam Tobin, Thomas Douglas, Akira Kato

**Advisor:** Professor Masoud Salehi

**Abstract**

Our team proposes an affordable, robust robot that can autonomously remove trash from beaches. As the beach becomes the focal point of marine pollution in the summer months, SANDBot will prevent trash from ever entering the water. With powerful motors, all-terrain wheels, precise GPS navigation, and computer vision-based collision avoidance, this robot can clean even the most hazardous of beaches. Multiple bots can be seamlessly integrated into a single fleet to clean larger beaches in a fraction of the time required by a team of human conservationists. By combining state-of-the-art technology with low-cost, efficient parts, SANDBot will end the war that mankind has been inadvertently waging on marine wildlife for decades. We looked at a few specific engineering design principles to concentrate our efforts and develop an optimal final product. First, we focused on sustainability - the main goal of our bot is to clean as much trash from the environment as possible. Furthermore, our team focused on usability - the robot needed to be simple to set up, empty out, program, and recharge so anyone would be able to take care of one with minimal difficulty. Finally, team members kept in mind the marketability of the final product - we wanted to make our device as cheap as possible yet extremely effective so any seaside community would want to purchase one.

To begin designing the model the first points of interest were the chassis and the collection tines since these two components determined the inner storage space for garbage and the necessary clearance for the robot. Once basic models for these components were created, a wheel mount that connects to the chassis and allows the robot to be closer to the ground was developed. The system is powered by a LiPo battery with full protection. During operation, the battery’s nominal voltage of 14.8 volts gets stepped down to 5V and 12V. The 5V rail powers the Raspberry Pi and the various logic level boards. The Raspberry Pi ended up being perfectly suited to the system, as we used every available GPIO pin and are projected to use an appreciable portion of its processing power, yet we didn’t have to limit the design to account for it. Based on our calculations, assuming a constant speed and rate of trash collection as well as frequent turning, the robot is projected to run for 2 hours and 20 minutes. In that time, it will travel a total of 11.26 km and collect 3.36 kg of trash, the equivalent of about 226 empty aluminum cans.

Our robot combines many complex systems. Computer vision, IMU processing, ultrasonic data collection, GPS information, a skid steering drive, localization, navigation, and a motor collection system. All of these discrete components needed to be connected in software, and in order to do so we are utilizing the ROS operating system. The benefit of ROS is that it maintains a set of pre-built modules that can be combined to take in external data from various systems and produce complex operations without having to produce excess code. ROS allows us to capture information from physical electrical components on a real robot, or from a simulated environment like Gazebo.

**EveRest Sleep Monitor**

**M3**: Ryan Hubelbank, Henry Jacobson, Priscilla Lin, Sean Sullivan, Daniel Vargas, Andy Wang

**Advisor:** Professor Masoud Salehi

**Abstract**

We present **EveRest**, a wearable sleep tracker which leverages machine learning to provide its users with advanced quantitative assessments of nightly sleep quality. EveRest is composed of three distinct components: a wristband featuring a broad array of physiological sensors, a state-of-the-art machine learning backend for analyzing sensor data, and a mobile application for reporting synthesized sleep analytics back to the user. EveRest aims to provide a near-medical grade accuracy in sleep tracking while being simple, convenient, and affordable.

Sleep disorders have been known to be immensely detrimental to an individual’s long-term health. However, current sleep tracking wearables unreliably measure sleep due to inadequate data collection (lack of sensor diversity) and insufficient data analysis (lack of AI competence). Conversely, medical sleep studies are feature-rich and accurate but prohibitively expensive and inconvenient. EveRest solves these problems with an abundance of purposefully selected sensors paired with an innovative two-layer neural network to conduct rigorous sleep analytics while remaining inexpensive and convenient. The total addressable market for EveRest is very large — as many as 70 million Americans actively suffer from chronic sleep disorders.

The EveRest wristband features 7 individual sensors (blood oxygen, heart rate, etc.), each sampling once per minute. Sensors are managed by a host microcontroller on a shared I2C bus and are explicitly chosen for impactful data yield and energy efficiency. Additionally, an on-board Bluetooth Low Energy (BLE) module streams sensor data to the user’s smartphone. This device is powered by a single 400mA•h LiPo battery and can operate for 9+ hours. The schematic, layout, and routing of the wristband PCB are completely custom designed. At scale, manufacturing costs are amortized to $31 per unit.

Data streamed from the wristband is uploaded to cloud storage and consumed by a dual-layer machine learning model. In the first layer, a support-vector machine (SVM) model converts sensor data to sleep metrics, such as sleep time and in-sleep movement. In the second layer, a least trimmed squares (LTS) regression model analyzes sleep metrics to produce a comprehensive sleep score. Additionally, tips and suggestions for improving sleep hygiene are also generated. Both models are trained using open-access data sets. The sleep score, sleep metrics, and suggestions are aggregated and presented to the user through a custom Android app.

Testing and validation have been conducted for both hardware and software components of EveRest. For hardware, functional verification is conducted in simulation and with real human subjects. Sensor data integrity is ensured by manually inspecting collected data and comparing the data to expected values. Finally, low power consumption of the wristband is confirmed through repeated battery life stress tests. To measure and optimize performance of the machine learning layers, portions of the training data are partitioned for testing. Each layer achieved a prediction accuracy of 85% and 91.9% respectively. From our repeated validation trials, we are confident that EveRest is a successful product.

**BeeMinder**

**Team M4:** Joseph Downing, Stephen Downing, Geralyn Moore, Zachary Weiss, Michael Busa

**Advisor:** Professor Masoud Salehi

**Abstract**

BeeMinder is a beehive monitoring system designed to help beekeepers easily and remotely monitor the health and well-being of their beehives. To achieve this, the BeeMinder system measures four aspects of a hive that reveal important information about the health of the hive: temperature, humidity, weight, and audio. The temperature, humidity, and audio are used to determine the health of the colony, while the weight is used to determine when the honey is ready to be harvested. Although there are similar products currently on the market, BeeMinder aims to improve on a number of the flaws in these systems. Specifically, the team created BeeMinder with three main goals in mind: keep it affordable, produce a low-power system, and design it to make the beekeepers’ lives as easy as possible.

The BeeMinder system consists of four main parts: the hardware system, communications, database, and user application. The hardware system is made up of two subsystems: the hive system (sensor module) and the base station. A beekeeper will have one hive system for each hive that they own, and one central base station that communicates with all the hives via Bluetooth Low Energy (BLE). The base station collects temperature, humidity, weight, and audio data from the hives and performs the data processing, including a Fast-Fourier Transform (FFT) on the audio data. The base station then sends the data to the MongoDB database. Finally, the user application, created using React and Node.js, displays the data and informs the beekeepers of the hives’ health. From the user application, beekeepers have the ability to view the temperature, humidity, and weight measurements, as well as inferences made based on the audio data and a graph of the FFT for further examination.

For testing, two hive systems were designed and fully constructed, each with a microcontroller, microphone, temperature and humidity sensor, load cells, battery, and solar panels. For the base station, the team used a Raspberry Pi 4 because it can use Bluetooth to communicate with the hive systems as well as Wifi to communicate with the database. With the full system implemented, data collected from the hive was processed by the base station, stored in the database and presented to the user in the webapp.

To view the full version of the user application, go to: <https://bee-minder.vercel.app>

The default login information should work, but if it does not, use the following to log in:

Username: Michael@beeminder.com

Password: 123bee