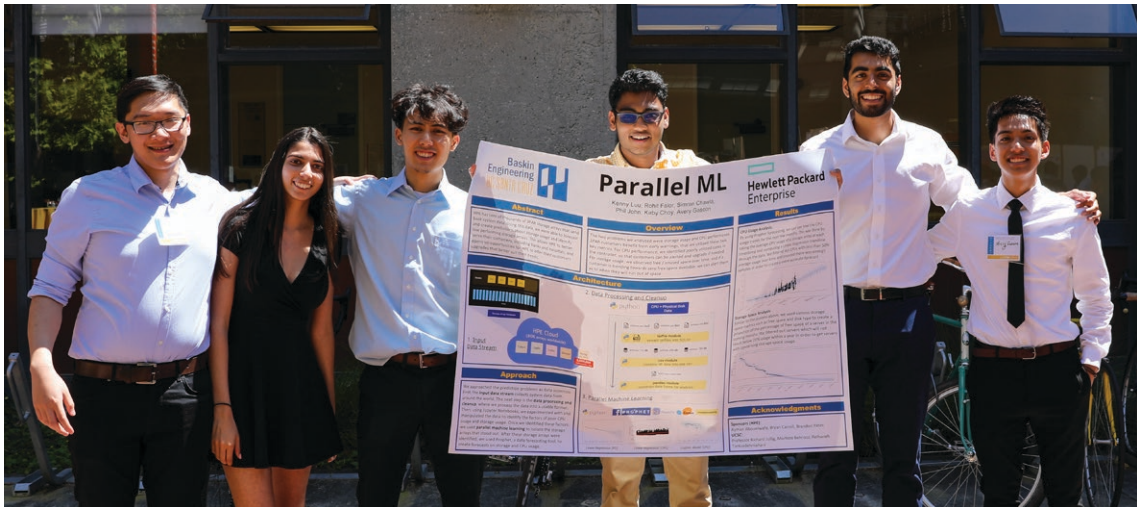
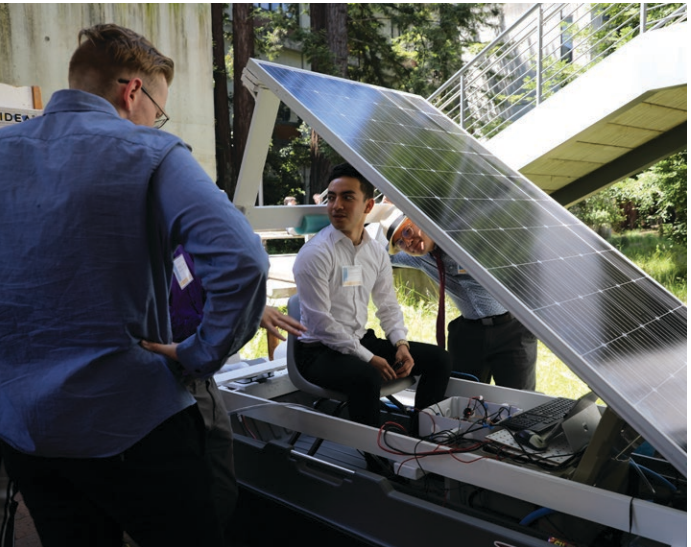
PARTNERS' DAY

Baskin Engineering UC SANTA CRUZ

2020-2021

CORPORATE SPONSORED SENIOR PROJECTS PROGRAM
& SENIOR DESIGN SHOWCASE





INTRODUCTION

This publication highlights the 10th year of the **Corporate Sponsored Senior Project Program (CSSPP)** at the UCSC Baskin School of Engineering.

CSSPP provides students with a unique opportunity to work on real-world engineering projects during their undergraduate education. Throughout the academic year, students interact with teammates and hold frequent meetings with their sponsors, getting feedback on the solutions they have developed and guidance on the work in progress. By working with mentors at corporate partner companies, students learn important skills, take on interesting challenges, and begin to understand what it means to be a professional engineer.

We appreciate our corporate sponsors for supporting CSSPP, mentoring our students, and providing them with challenging projects to work on. We also appreciate our students, who have worked hard and have enriched our lives through their energy, intellect, and determination. This year has featured significant disruptions, as both students and sponsors have had to interact in a distributed environment. Students worked remotely, and even those living locally had no access to the UCSC campus. All of our students – many of whom worked from locations around the globe – had to be tremendously resourceful to find ways to work together productively and advance their projects.

This publication also includes this year's Senior Design Program Projects from student teams in Computer Science & Engineering and Electrical & Computer Engineering who worked on faculty/student initiated projects. As with our CSSPP projects, students working on our senior design program had to be agile and creative in pursuing their projects to completion in the face of limited access to critical resources and to one another.

We appreciate the flexibility and creativity of our corporate partners, our faculty mentors, and our students as they have worked through this challenging time, and acknowledge that with industry evolving in response to employees' growing adoption of remote work, the experience our seniors have had this year will likely prepare them well for what is on the horizon professionally.

Alexander L. Wolf
Dean
Baskin School of Engineering

Baskin Engineering UC SANTA CRUZ

ACKNOWLEDGMENTS

We would like to acknowledge and thank the faculty, teaching assistants and staff who have been so instrumental in the Corporate Sponsored Senior Projects Program:

SENIOR DESIGN FACULTY
CORPORATE SPONSORED
SENIOR PROJECTS PROGRAM 2020-21

Patrick Mantey
Director, Senior Design Capstone, Jack Baskin Endowed Professor, Computer Engineering, Emeritus, Baskin School of Engineering

Richard Jullig
Lecturer, Electrical & Computer Engineering

Tela Favaloro
Lecturer, Electrical & Computer Engineering

David Harrison
Lecturer, Computer Science & Engineering

Stephen Petersen
Teaching Professor, Electrical & Computer Engineering

TEACHING ASSISTANTS & GRADUATE RESEARCH ASSISTANTS
CORPORATE SPONSORED
SENIOR PROJECTS PROGRAM 2020-21

Faeze Brahman

Tanner Miller

Neil Hardy

Sushmita Joardar

Azzam Qureshi

Baskin Engineering UC SANTA CRUZ

CORPORATE SPONSORED SENIOR PROJECTS PROGRAM STAFF

Alexander Wolf
Dean, Baskin School of Engineering

Frank Howley
Senior Director of Corporate Development, Baskin School of Engineering

Peter Minogue
Director of Laboratories and Facilities, Baskin School of Engineering

Richard Sabala
Director of Laboratories & Facilities, Baskin School of Engineering

Russell Evans
Senior Development Engineer, Baskin School of Engineering

Devin Gayne Riley
BELS Teaching Lab Engineer

Jeffrey Jue
Special Agreements Officer, Industry Alliances & Technology Commercialization

Kelsey Evans Dewiler
Development Assistant, Baskin School of Engineering, University Relations

Stacy Wilson
Executive Assistant to the Dean, Baskin School of Engineering

Ady Shapira Weinstein
Event Coordinator, Baskin School of Engineering

Melissa Rochelle Baker
Event Coordinator, Baskin School of Engineering

Alexander Cosmo Martinez
Web Developer, Baskin School of Engineering

SPONSORS

Special thanks to our sponsors for your generous support of our Corporate Sponsored Senior Projects Program. Your time, experience and financial support were beneficial to our students and the success of their Senior Design Projects.

UC SANTA CRUZ
Baskin Engineering

Andy Gordon
andgordo@ucsc.edu
Aidan Smith
aidansmith192@gmail.com

Capstone Project

GoGuard

Rory Landau
rflandau@pm.me
Surya Suresh
ssuresh3@ucsc.edu

CISCO

Abstract

GoGuard is a cure to the messy menagerie of technologies in established private clouds. It builds and manages a transparent, secure, homogeneous network of services for a private cloud.

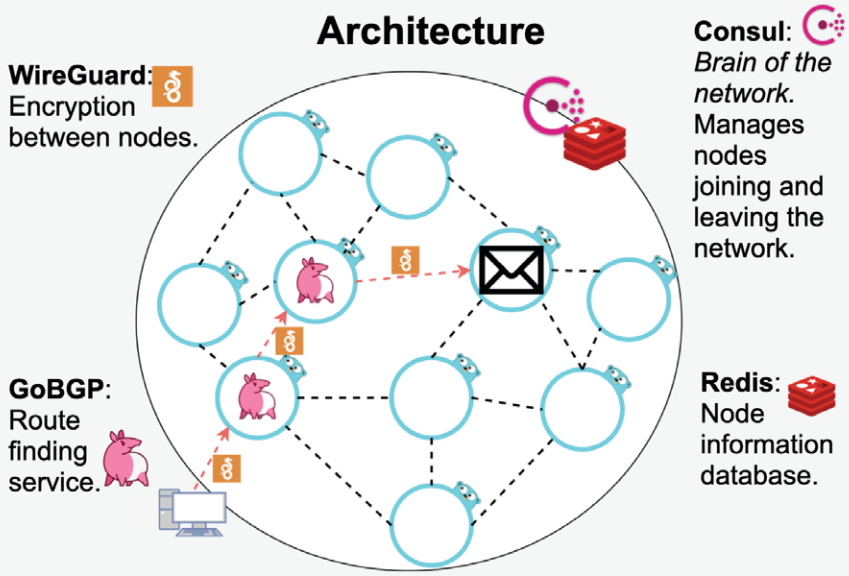
Approach

1. Identify technologies that…
   manage networks: **Consul**
   build secure tunnels: **Wireguard**
   store network info: **Redis**
   and *route traffic*: **BGP**
2. Bootstrap a network skeleton using Consul, Wireguard, Redis
3. Connect services via tunnels to build a mesh network. Consul aids in adding/removing services from this network.
4. Improve network reliability via Border Gateway Protocol
5. Package GoGuard for portability and production

Overview

CISCO's current solution to manage Umbrella services and client connections is through a heterogeneous network running different VPN technologies. Our goal was to create a homogeneous private network where services connect and communicate via Wireguard and are informed of changes in the network by Consul.

Architecture



Connecting Service to the GoGuard Network

1. Service authenticates with Consul
2. Consul registers service
3. Connected!
4. Service now has full access to mesh routing and other nodes/services via encrypted channels

Results

1. Created a private, data-agnostic network builder and manager
2. Built Wireguard connections between services
3. Used dynamic routing of traffic
4. Stored and managed peering information of services locally and via Redis and Consul
5. Packaged using Docker for ease of use in development

Conclusion

GoGuard is *scalable and extensible*; it can theoretically support any type of service or client and is ready for deployment. Future work could improve routing capabilities and provide a hook for local applications looking to access GoGuard's information.

Acknowledgments

Thanks to the CISCO team:
Kyle Mestery, Alex Dickinson, Adrian Oliver, Adam Brown, and Sanjeet Jain.
And to our teaching staff:
Richard Jullig & Faeze Brahman

UC SANTA CRUZ
Baskin Engineering

Capstone Project

Spark IP Address API

Julian Shalaby, Ryan Jacobs, Prateek Narayanan

databricks

Abstract

SparkIP is an API that adds support for IP addresses to Apache Spark. The API is designed to be intuitive and efficient, allowing users to manipulate data containing IP addresses. This is particularly relevant to security teams who need a way of querying millions of network logs with efficiency and ease.

Approach

We implemented three fundamental data structures which enable users to work with IP data.

IPAddress: Either an IPv4 or IPv6 address. Forms the basis of IPNetwork.

IPNetwork: Represents a contiguous sequence of IPv4 or IPv6 addresses.

IPSet: An arbitrary set of IP addresses and/or IP networks.

The above data structures are baked into Spark using both User Defined Types (UDTs) to represent the data types, and User Defined Functions (UDFs) for functionality using the data types.

Overview

Databricks is a data and AI company founded by the creators of Apache Spark, an open source platform used for large scale data processing. Since security teams need to work with IP data at scale to effectively identify threats in their network, Spark is a great choice for their tasks. Due to Spark not providing native support for IP data, many of these complex tasks can become difficult to perform. For this reason, we made SparkIP, an API to enable handling IP data natively in Spark.

Code Comparisons

Match all IP addresses within a given network

Pattern matching (current practice):

```
SELECT * FROM log WHERE REGEXP_MATCH("IPAddress", "^192\.0\.([0-9]|
[0-9][0-9]|1[0-1][0-9]|12[0-7])\.[0-9]+$")
```

SparkIP:

```
SELECT * FROM log WHERE networkContains("IPAddress", "192.0.0.0/17")
```

Match all IPs in either of these networks: 192.0.2.0/26 or 10.0.1.0/24

Pattern matching:

```
SELECT * FROM log WHERE REGEXP_MATCH("IPAddress", "^192\.0\.2\.([0-
9]|[1-5][0-9]|6[0-3])|10\.0\.1\.([0-9]|1[0-9][0-9])$")
```

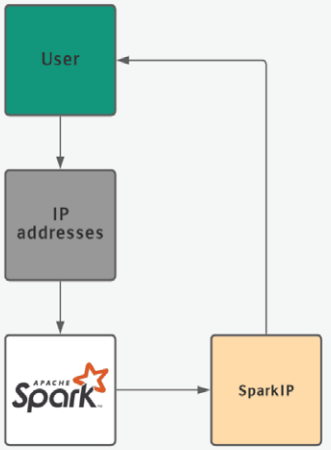
SparkIP:

```
val set1 = IPSet(["192.0.2.0/26", "10.0.1.0/24"])
SELECT * FROM log WHERE setContains("IPAddress", "set1")
```

Special Thanks to Our:

Sponsors: Markus De Shon, Matt Yang, David Veuve
Professor: Richard Jullig
TA: Faeze Brahman

Architecture



Results

- SparkIP: Comparable in speed to Spark's current pattern matching approach
- Published PySparkIP on *PyPi*
  - Installable with *pip*
- Published SparkIP
  - Importable in Scala

Conclusion

Using the API we created, security teams will be able to handle IP addresses with ease in Spark, which in turn, could help improve the security of the team's platform.

UC SANTA CRUZ
Baskin Engineering

Capstone Project

Dell Cloud Garbage Collection

Luke Harvey, Michael Hardy, Herbert Li,
Sanyam Sachdeva, Vidisha Nevatia

DELL

Abstract

The goal of this project was to explore cheaper ways of removing garbage data from cloud based backup systems. In order to help reach this goal we tested ways of removing garbage data for the data protection division of Dell EMC. In the course of testing we found that using Lambda services to offload CPU intensive tasks from the in-cloud data protection system scaled well, but in one section was bottlenecked by DynamoDB I/O.

Approach

Files are split into fixed sized segments; deduplication removes duplicate segments to reduce storage size. The segments are stored in containers in Amazon S3. The files and their segment structure are stored in DynamoDB.

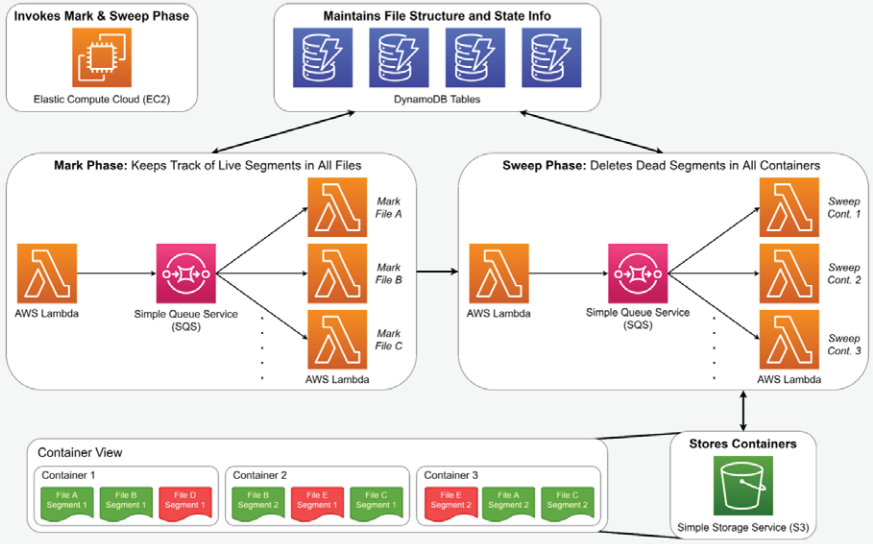
As directed, we use the *Mark and Sweep* methodology for garbage collection (GC).
The **Mark Phase** iterates over the files in the system in parallel lambda instances, **mark**ing all the live segments as *used* in a bloom filter (a space efficient probabilistic data structure that records which segments are probably used). This filter is stored in DynamoDB.
The **Sweep Phase** (also parallelized) inspects the bloom filter to copy marked segments forward into new containers and delete old ones.

We created a **benchmarking suite** to vary the degree of parallelization of both mark and sweep, and vary the size and amount of segments and containers. We also created a set of scripts to **automate the setup** of our infrastructure in AWS.

Overview

Our goal was to analyze the efficiency and cost effectiveness of moving **garbage collection** from virtual machines to cloud functions in cloud-based data protection solutions. Traditional physical data protection solutions come with enough CPU resources such that they are capable of doing the most CPU intensive task (garbage collection). Cloud-based systems need not always have CPU allocated for intermittent tasks like garbage collection and can instead leverage short task systems like AWS Lambda.

Cloud Garbage Collection



Acknowledgments

We would like to thank Smriti Thakkar, George Mathew, and Mansur Khan of Dell Technologies, professor Richard Jullig, and teaching assistant Faeze Brahman for all of their encouragement and support throughout the project.

Glossary

Garbage Collection - The process of removing unneeded data from memory or storage systems
Deduplication - Removal of duplicated sections of data
Key value store - A database storing a set of values by key. For example a set of medical records indexed by name.

AWS Services:
Lambda - Computing service that runs small code snippets on demand
EC2 - Rental computers in the cloud
DynamoDB - A Key Value store Database
S3 - An Object Store Database
SQS - Message queuing system used to wake up lambda nodes

Results

- As the **Sweep Phase** is parallelized over more lambda instances, we saw lowered cost and faster runtimes.
- The **Mark Phase**'s I/O dependent workload only scales partially, and is bottlenecked by limitations with DynamoDB.

Deliverables

- **Highly modular Benchmarking Suite** that allows for fast and efficient benchmarking.
- **Set of CloudFormation scripts** to automatically set up the benchmarking infrastructure
- **Collected Benchmarking Statistics** that show the effect of parallelization of mark and sweep

Capstone Project

PSLab Integration for OpenTAP

Alex Kostin, Ivan Diep

Abstract

The integration of the PSLab board as a plugin in Keysight's OpenTAP software allows for the measurements done with PSLab to be automatically collected and viewed in a series of test steps within OpenTap. This allows for ease of use for students trying to learn electrical measurement concepts using PSLab. We built a course around this idea so that students trying to learn these properties can have a more interactive experience.

Technologies

Software: OpenTAP is an open source test automation software that performs testing tasks in a straightforward and repeatable manner. It only comes with a test sequencer and a command line interface, but provides access to additional functionality with plugins. Adding plugins to OpenTAP allow for more features such as a GUI, results visualization, and many more.

Hardware: The PSLab board is an open source, portable hardware board that can be used to make measurements with the many different onboard instruments it provides.

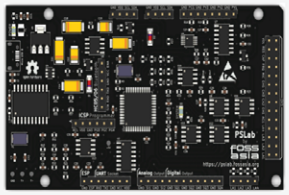
Approach

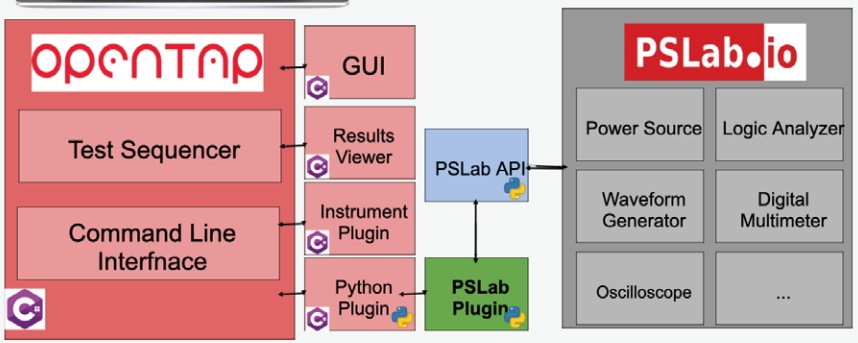
We created a **PSLab plugin** that allows interaction between OpenTAP and PSLab. The plugin calls Python-written methods from the PSLab API to interact with the PSLab onboard instruments. Our plugin relies on OpenTAP's Python plugin to translate our Python code into C# code readable by OpenTAP's C# API.

Overview

Testing Automation is the process of taking manual testing tasks, like seeing if a current between two points is the expected amount or if clicking a button in your program does the expected thing, and having a program run and report on those tasks for you. Our project helped Keysight's OpenTAP Testing Automation program use the PSLab measurement board to do testing tasks like measuring voltage, capacitance, and more! We also created a course to help students use OpenTAP + PSLab to learn EE-related concepts.

Architecture





Acknowledgments

We would like to thank our Keysight sponsors: **Brennen DiRenzo**, **Kaushik Santhanam**, and **Jeff Dralla**, our professor **Richard Jullig** and TA **Faeze Brahman,** and our former team member **Ashton Humphrey** for helping this project come to fruition.

Course Content

Our course will contain multiple labs that implement our PSLab plugin for OpenTAP. It will teach students EE concepts using the instruments provided by PSLab. Our course will also teach students how to build their own plugins in Python using OpenTAP and PSLab's API. The course is made accessible even to students that do not own the PSLab board through simplified simulated instruments that are built into our PSLab plugin.

Results

PSLab Plugin

- Created access to all instruments provided by PSLab API:
  - Power Supply
  - Wave Generator
  - Oscilloscope
  - Logic Analyzer
  - Digital Multimeter
- Created 24 related test steps to interact with these instruments, such as:
  - Creating a waveform
  - Changing voltage/current
  - Measuring capacitance/resistance

Course

- Created a course that allows students to learn about EE concepts and how to build OpenTAP plugins.

Conclusion

Open sourcing our PSLab plugin allows anyone to build upon our code. The extensibility of OpenTAP plugins and PSLab's continued development allows our project to always stay up-to-date. Our labs will teach students test automation and building plugins with OpenTap + PSLab.

UC SANTA CRUZ
Baskin Engineering

Capstone Project

Ransomware Detection

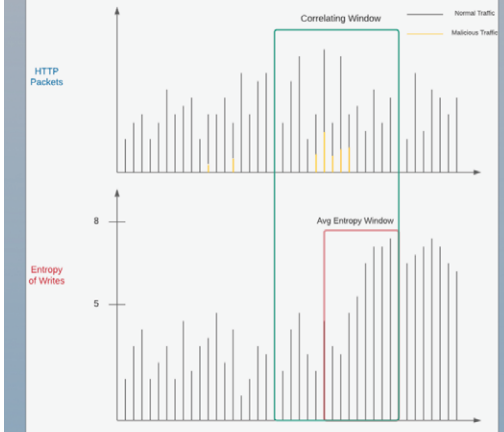
Scott Fischer, Raphael Zaafrani, Cyrus Karsan

NUTANIX

Abstract

Our ransomware attack detection system is designed for companies running private cloud servers much like Nutanix's clients. Our system has 2 main layers of detection to ensure a high accuracy. We use both machine learning and rule-based algorithms to ensure the accuracy for the attack alarm.

Correlating Signals

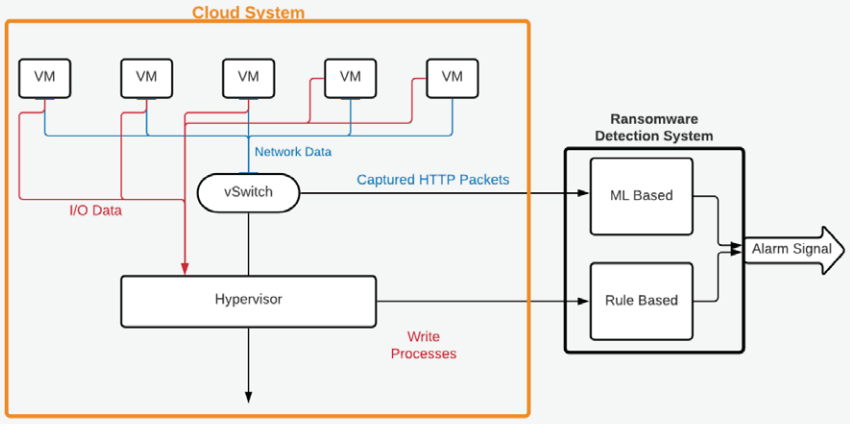


In order to use both network and storage encryption detection we use a sliding window that constantly analyzes and labels the output of our ML and rule-based algorithms. When it sees a malicious network label followed by several high entropy labels it sounds the alarm.

Overview

Ransomware is malware designed to encrypt files on a machine, essentially locking that system and charging a ransom for decryption key. Our goal was to (1) research signals that indicate a ransomware attack and (2) to design a system to detect ransomware attacks before a full lock down. Our system is designed to work on cloud systems that use a hypervisor to manage the many VMs in the cloud system.

Architecture



Acknowledgments

Big thanks to Nutanix and Bharat Beedu for sponsoring this project!
Thanks to Professor Richard Jullig, Faeze, and the UCSC CS department for supporting this project.

Analysis

Our decision tree gave us an accuracy of about 93.7% with an F1 score of 0.94, and an average precision on all labels of 0.945. The F1 score is the weighted average of precision and recall.

With entropy we will have some false positives with compressed data that is written but using the network data as a trigger eliminates this worry.

Results

- Created a list of signals that ransomware emit that can be used for detection
- Created a system with two levels of detection
- Integrated rule-based algorithms with machine learning to increase confidence

Conclusion

If the project were to be worked on further, we would suggest adding signals such as CPU usage, CPU opcodes, and memory write locations. Adding these signals would strengthen the system and would increase the accuracy of the alert.

UC SANTA CRUZ
Baskin Engineering

Capstone Project

NVMe-OF Characterization

Steven Resendiz, Michael Wang, Dylan Yong, Christopher Calderon, Aaron Nguyen

SAP

Abstract

Storage products have quickly evolved to enable data transfer at incredible speeds with a rapid focus on improving latency and cpu usage.

Based on this need, the project focused on creating a comparison between different network protocols. This comparison was built using software tools measuring network performance.

SAP is utilizing this evaluation to transition their servers to become more efficient. This uptick in performance provides faster processing of data and enables the server to handle more requests in less time.

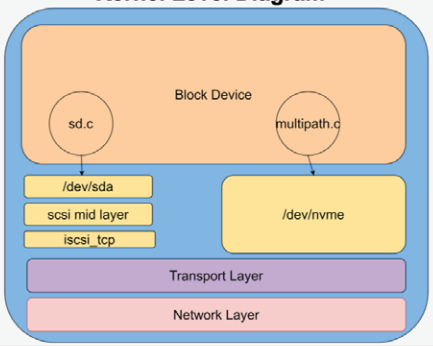
Methods

Measurements for latency and bandwidth were taken using the flexible I/O tool (FIO). To measure latency, this tool determines the round trip time between entering a layer and resolving requests.

Bandwidth and iops are measurements of throughput. Bandwidth being concerned with volume of data over time, versus iops which is concerned with number of io operations over time.

Initial approach consisted of kernel level timings added to sd.c and multipath.c module to measure latency between each kernel layer (block, transport, network) as shown below.

Kernel Level Diagram



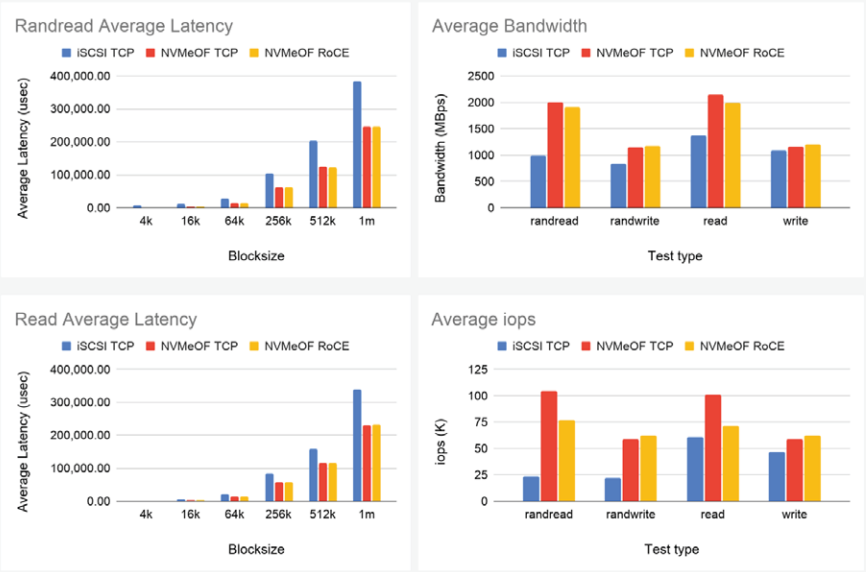
Overview

SAP Global Cloud Service(GCS) is the division of SAP focused on providing cloud services and infrastructure, including IaaS/PaaS solutions, such as HANA cloud services, and SaaS solutions, such as Concur and Ariba.

In modern day storage, I/O bottlenecks have shifted from storage hardware performance to transport protocols. To take advantage of these advancements, several storage transport protocols have been developed to minimize this I/O bottleneck.

In our project, we investigated different storage transport protocols to determine which would be best for SAP systems based on gathered latency and bandwidth data. Resulting evaluations could guide SAP in improving the performance of their cloud servers, and reducing cost.

NVMe-OF and iSCSI Measurements



Analysis

The network measurements we performed were of bandwidth and latency. Each test performed one of four operations: sequential read, random read, sequential write, and random read. Included are the results of read/randread latency measurements and bandwidth/iops measurements. For bandwidth measurements, a higher MBps is better and we see that NVMeOF - TCP is reaching scores near twice that of iSCSI. For latency measurements, a smaller average is better. We see that NVMe-OF consistently reaches lower latencies than iSCSI using TCP. This is true for read and write measurements, although it is far greater for read operations. For iops, a larger number is better and we see NVMe-OF out-performing iSCSI.

Definitions

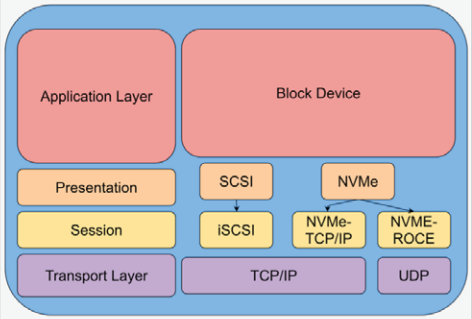
NVMe-OF (Non-Volatile Memory Express over Fabric): Network Storage Protocol that uitilizes Network protocols to allow for remote storage over a fabric. This protocol is designed for NVMe, allowing for the creation of 64k IO queues.

iSCSI (Internet Small Computers Interface): Network storage protocol that utilizes SCSI protocol to connect target/initiator system. iSCSI creates less IO queues than NVMe, yet it is a more established technology in the industry.

TCP/IP (Transmission Control Protocol/Internet Protocol): Network protocol that sends/receives packets. TCP establishes an end-to-end connection, which ensures packets are transmitted successfully.

RoCE (RDMA over converged Ethernet): Network protocol that utilizes UDP instead of TCP for transport. UDP differs from TCP in that it lacks an end-to-end connection and packet checking. This means that UDP is vulnerable to packet loss if the network is unreliable.

OSI Model



Acknowledgments

Thank you to:
Jody Glider (Sponsor)
Raghu Govindasamy (Mentor)
Professor Richard Jullig
Faeze Brahman(TA advisor)

We are very pleased to include posters for the Senior Design Projects that were done without industry sponsors. Some of these projects were instigated and/or sponsored by research at the Baskin School while others were created by students with the assistance of faculty mentors and TAs.

Extended Flight Time Drone

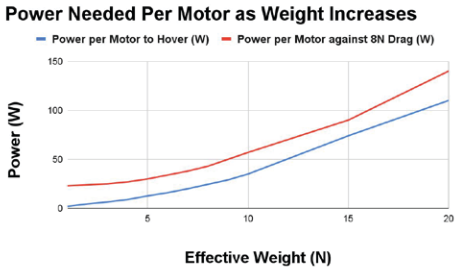
Dylan Harootunian, Chin Ming Ryan Wong, Leonid Shuster, Jeremy Germenis, George Hernandez, Isaac Szu
Electrical and Computer Engineering Department

Abstract

The United States Geological Survey uses drones to collect magnetometer data, but they are limited to 15 minute flights. Counteracting a drone's weight consumes a majority of its onboard battery power. We have designed "The Barone" to incorporate buoyancy to reduce its own effective weight and increase flight time to 40 minutes at 20 mph air speed, and up to 100 minutes while hovering.

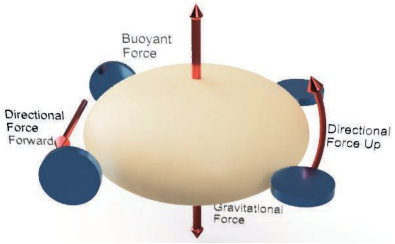
Problem

A drone spends most of its energy for the propulsion system to counteract the system weight, resulting in shorter flight times.



Approach

By using a helium lift bag, the force of buoyancy can work against gravity, and reduce the "**effective weight**;" the motors would need less energy to maintain altitude, increasing drone flight time. Due to the buoyant moment, the drone cannot tilt like standard quadcopters, so a custom propulsion system is needed.

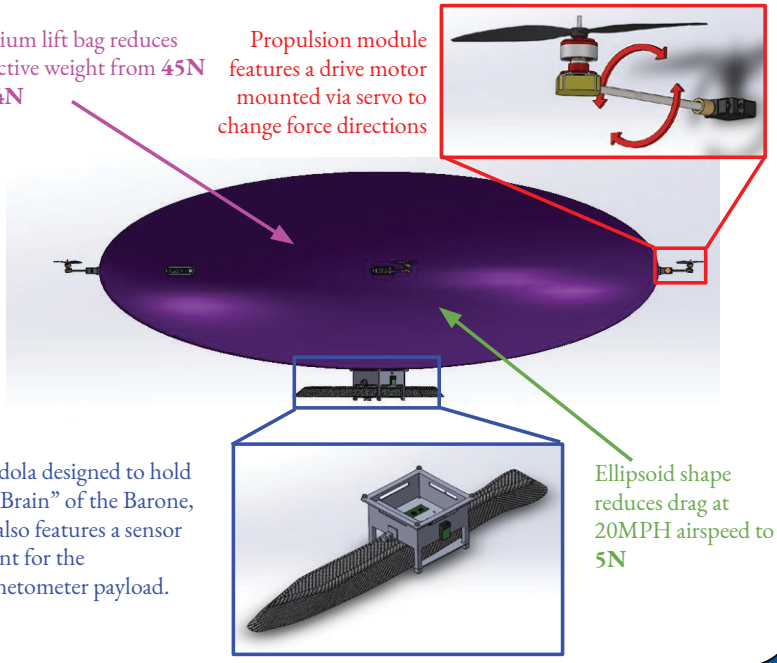


The Barone, Early Force Diagram

The Barone

Helium lift bag reduces effective weight from **45N to 4N**

Propulsion module features a drive motor mounted via servo to change force directions



Gondola designed to hold the "Brain" of the Barone, and also features a sensor mount for the magnetometer payload.

Ellipsoid shape reduces drag at 20MPH airspeed to **5N**

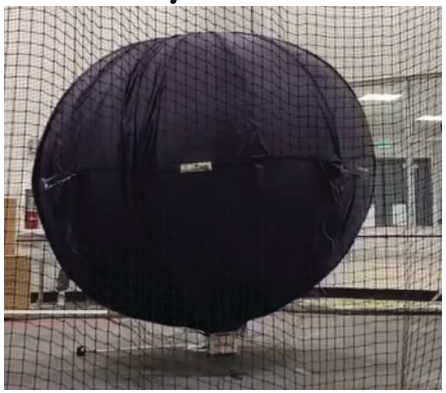
Stable Tilt

Most drones need to tilt to maneuver, but using the drone's buoyant moment as well as its custom propulsion system, the tilt angle of the drone is kept near zero, providing a more stable payload platform than other drones.

Acknowledgments

We would like to thank Jonathan Glenn and the USGS for working with us to develop our drone requirements, Professors Stephen Peterson, Tela Favaloro, and Mircea Teodorescu, as well as graduate students Gordon Keller, Alexey Munishkin, and Tanner Miller for their guidance, and Porter and Crown College for their financial support.

Physical Test



- Confirmed power draw of parts to reach **40-100 minute flight**
- Confirmed system weight < **7N**
- Envelope failed to hold propulsion module outright when not fully inflated
- Helium loss test revealed a loss of **8.4% an hour**

Conclusions

- The Barone is a proof of concept for a long flight time buoyant drone design.
- More clients are needed to adjust The Barone to meet a variety of user needs.
- Aerodynamic optimization is needed to extend flight time.
- Manufacturing process needs to be refined to achieve tighter dimensional tolerances.

Automatic Facial Recognition Door

Gursewak Sokhi, Sanjeevram Duraivelu, Ryanul Haque, Aravind Chinthada, Vuong Do, Israel Coronado, Noah House

Team 3

INTRODUCTION

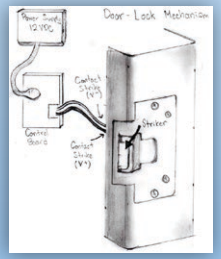
Need Statement:
People forget their keys and lock themselves outside.

Goal Statement:
To make a device that would be able to unlock without a key.

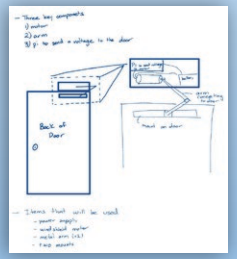
Personas:
- Forgetful people who tend to leave their keys
- Homeowners who are looking for better and easily accessible security

Design Objectives:
- Inexpensive locking system that will unlock your door with facial recognition
- Camera that will act as a sensor to detect the face and unlock the door
- Easy installation and low power consumption
- Completely hand-free product

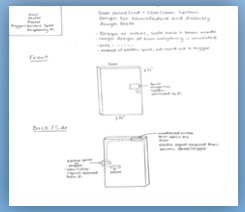
DESIGN FOR MANUFACTURING



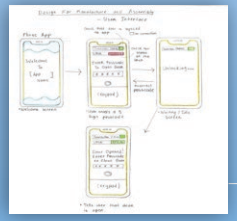
Electric strike and door locking mechanism



Motor connection to the door

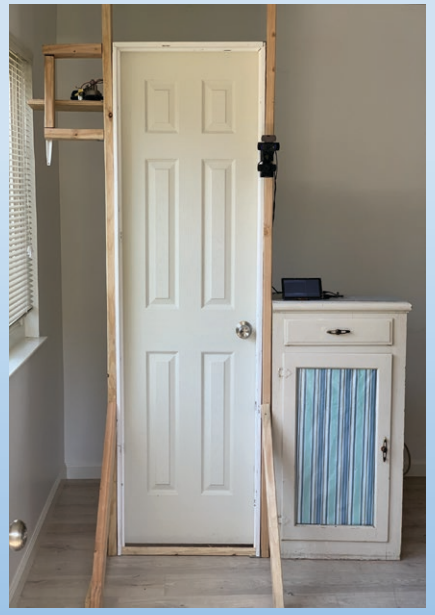


Full overview of door, motor, and electric strike lock



App to open the door

FINAL PROTOTYPE DESIGN

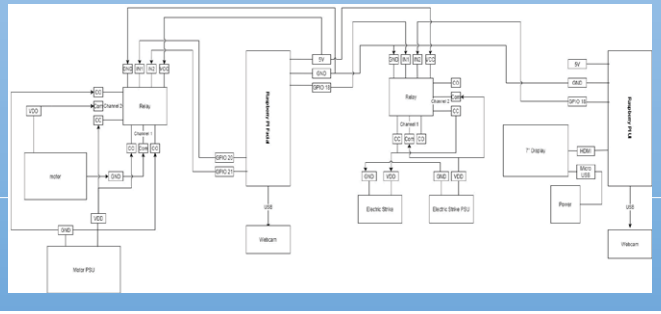


Front view of functional prototype

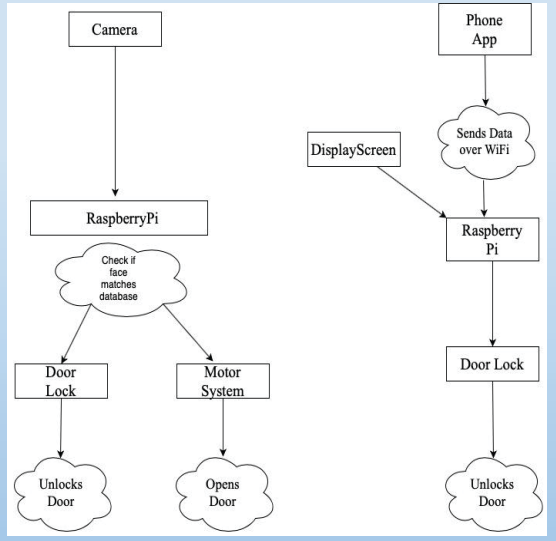
App interface using Flutter

Display showing real time camera feed

WIRING DIAGRAM



OVERALL DESIGN FLOW CHART



- A RasPi is used for the facial recognition and the mechanical locking system.
- Once the camera detects a valid face it sends a signal to the electric strike which unlocks the door
- It also sends signal to the motor which opens the door.
- The pi on the right was used for the user interface and the phone app.
- The display is connected to the pi.
- The phone app is used to send a signal to the electric strike to unlock the door

BRANDING

Baskin School Of Engineering
University of California, Santa Cruz

ACKNOWLEDGEMENTS

Thank you to the BSOE, Professor Harrison, and the TA, Kiefer Selmon, for guiding us through this project .

Affordable Piano Amplification

Capstone Project by **Evan Collins-Smith, Eric Liu, Evan Mercado, Abdul Nomair**

BACKGROUND

Pianos need to be amplified in ensembles as they are quieter than other instruments, but the standard processes for amplifying pianos, especially at the high school level, have major problems:

1. Microphones pick up other instruments besides just the piano
2. The *Helpinstill* electromagnetic pickup system for pianos has a minimum price of $700.

High school music programs are looking for a piano amplifier that picks up less noise than a microphone, and that is more affordable than the *Helpinstill* system.

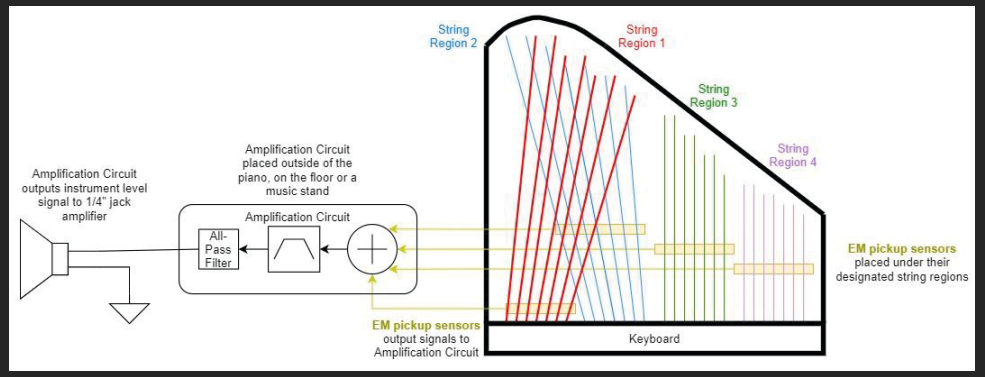
OBJECTIVE

Our system aims to reproduce a piano's sound wave as an electrical signal that can be read by an amplifier with a cost between $250-350.

APPROACH

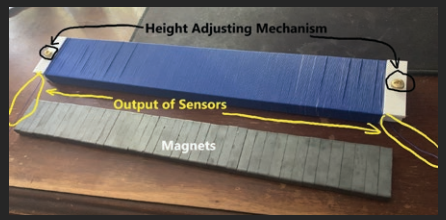
Our system can be divided into two main components: Electromagnetic Sensors and an Amplification Circuit. The electromagnetic sensors, placed under the strings, reproduce the piano's string movement through electromagnetic induction. The signals produced by these sensors are sent to the Amplification Circuit. The Amplification Circuit adds the four signals and amplifies them, outputting their sum at instrument level. This output is sent to a ¼" jack amplifier.

Take a picture to download the full paper



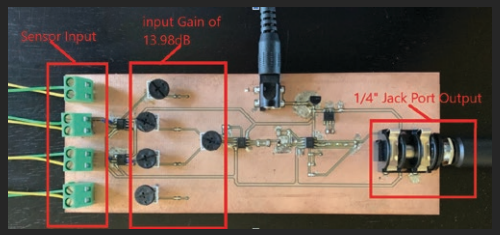
System Overview: Sensors are placed on the soundboard of the piano, under their designated string regions. Wires connect the sensors to the Amplification Circuit. The Amplification Circuit is placed outside of the piano, powered by a wall outlet, and outputs the resultant signal to a ¼" jack amplifier.

Electromagnetic Sensor



An electric signal is produced at the ends of the sensor when a piano string vibrates. The sensors are height adjustable to get as close to the strings as possible without touching them.

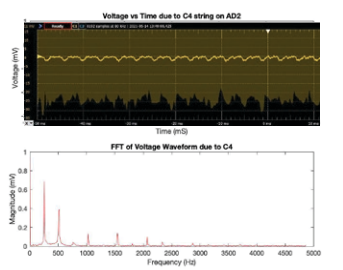
Amplification Circuit



Four input signals are summed together, amplified, and passed through a bandpass filter and an all-pass filter.

RESULTS

Signal measured from prototype sensor under the note C4



Measured voltage was a little over $4mV_{pp}$. The fundamental frequency is in the correct location, and the magnitudes of the overtones match professional recordings

CONCLUSION

Testing with the prototype sensor closely resembled simulation. A 2mV offset was discovered, but the believed cause of this discrepancy is measuring error. Further investigation is required.

Testing with the circuit was impeded by heavy noise. The cause of this noise was attributed to fluctuations in the negative power supply. This issue can be fixed by using a different capacitor for the voltage converter.

SimpleSort

Justyn Duthler, Kevin Ryan, Shirley Phuong, Trevor Carleton, Kevin Spence, Brevan Chun
CSE 123 2021

Project Overview

Need: Too many recyclable and compostable items are being placed in the trash.

Goal: Increase the recycling and composting rates while decreasing the rate of garbage improperly sorted.

Design Objectives:

- Automated home use garbage sorter/ garbage can
- Three compartments: trash, compost, recycling
- Image classification to determine which compartment the placed object belongs in
- Sorting mechanism that rotates and drops object into correct compartment
- > 80% Sorting accuracy for proof of concept

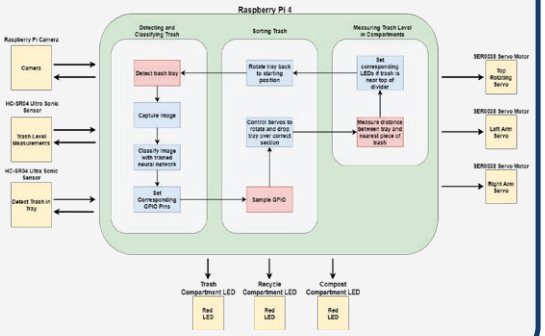
This unit would be intended for at home use, reducing the sorting required at larger facilities. We prioritized a simple design and low cost for manufacturing to make it more accessible for consumers.

Target Market:

Luna Rosewood :
Environmentally conscious trash producer

Martin DeMaria :
Busy homeowner, does not have time to recycle or compost

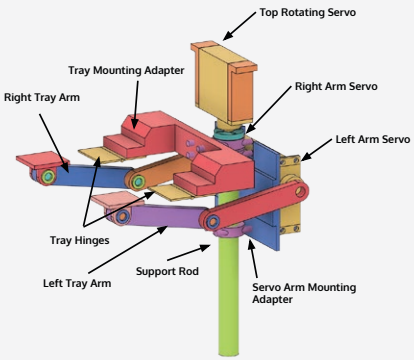
Architecture:

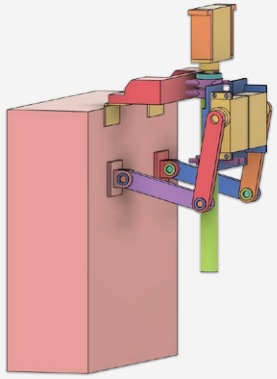


Design

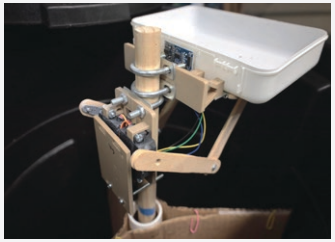
Aesthetic Rendering:

Sorting Mechanism:





Physical Prototype:



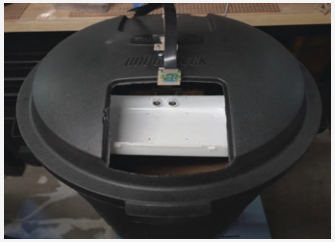
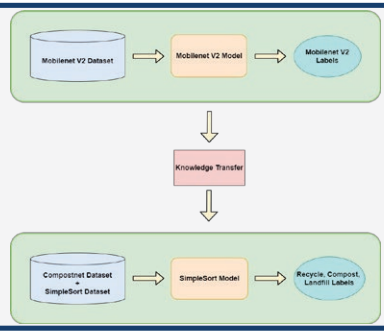
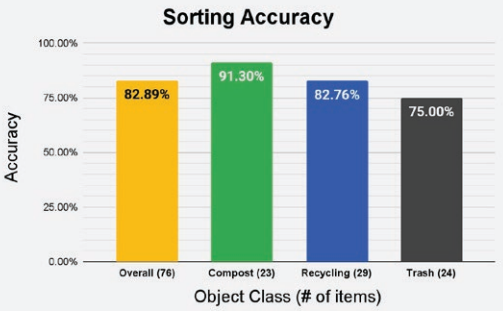


Image Classification



Results



Conclusion:

As seen in the graph above, we were able to surpass our accuracy goal of 80%. The compost class was the most successful at 91.3% and the trash class had the lowest accuracy at 75%. This is likely due to the disparity between the compost subset and the trash subset sizes in our dataset. Given more time, we would increase the size of the trash and recycling subsets to solve this inconsistency. On the same note, we would train our model to include more uncommon items such as compostable trash bags. In order to create a more consumer-friendly product, we would also compact the sorting mechanism to allow for more space in the can.

Acknowledgements

We would like to thank Professor Harrison and Kiefer Selmon for the support and knowledge throughout the design process. We would also like to thank BELS for supplying us with the hardware necessary to complete our prototype.

Contact Information: simplesort2021@gmail.com

PARTNERS' DAY
10
YEAR
ANNIVERSARY

CONTACT US

Patrick Mantey
Director, Senior Design Capstone
Jack Baskin Endowed Professor,
Computer Engineering, Emeritus
mantey@ucsc.edu

Frank Howley
Senior Director of Corporate Development
fhowley@ucsc.edu

Visit our website:
csspp.soe.ucsc.edu

Stay Connected:
LinkedIn: Jack Baskin School of Engineering
Twitter: UCSC_BSOE
Facebook: BaskinSchoolofEngineering
Instagram: UCSCengineering
YouTube: Jack Baskin School of Engineering

Baskin Engineering UC SANTA CRUZ