Autonomous Acoustic Deterrence System for Agricultural Use A.A.D.S

NEW YORK INSTITUTE OF TECHNOLOGY

Course: EENG 491

Professor: Aydin Farajidavar

Written By:

Rasheed Martin

Daniel Swarns

Hussein-Ali Mourtada

Constantine Strati

Table of Contents

Abstract **3**
Introduction **3**
System Design **6**
Structural Drawings 6
Final System Design 8
Electrical Schematic 9
Components 10
Power Consumption 10
Methodology **11**
Block Diagram 11
Software Design **12**
Flow Chart 12
Computer Vision with Deep Learning 14
Background 14
Implementation 15
Testing Data / Results **16**
Accuracy of Image Recognition Algorithm 16
Conclusion **17**
Limitations 17
Future Enhancements 18
Learning Outcomes 19
Bibliography **20**

Abstract

Gardening is popular in the United States of America, with roughly 55% of Americans taking part in the activity. Many people enjoy planting fruit trees and crops in their gardens. However, the major problems that come with gardening are the animals that eat the crops; sometimes even before it fully develops. Scarecrows are the #1 animal deterrents across the country, but they are not fully effective. Different animals eventually get used to seeing the scarecrow and stop running away from it. Our system aims to enhance the classic design of a scarecrow with additional features that improve its effectiveness in deterring animals. The system will begin by conducting the image detection algorithm and check for one of the following: racoon, bird, or squirrel. If it is the case that one of these animals is detected, an ultrasonic sound wave will be emitted from an attached speaker which will scare away the animal that was detected.

Introduction

Agriculture is a fundamental practice that impacts every society. The need for consistent, quality food production is clear. However, around 10-16% of global crop production is lost to pests **[13].** Pests include, but are not limited to, the following: insects, bacteria, fungi, birds, and rodents. The proposed system will be targeting the larger and more identifiable pests: squirrels, racoons, and birds.

One study shows that pests cause up to $220 billion of losses **[14]** and the problem is only getting worse. Pest control has been getting more difficult because of climate change and other serious environmental challenges. The warmer temperatures globally have altered the migration patterns of many species, causing the establishment of invasive pests. All it takes is one unusually warm winter in a cooler climate for this to happen. **[14]**

If pest control is connected to global warming, then it is a problem that is only going to get worse. One current method of mitigating this problem is focusing more attention on climate change strategies. This includes raising awareness and collaboration with farmers who live in impacted regions. Climate change induced pests cause the biggest problems in cooler climates such as the arctic, boreal, and subtropical regions.

Another angle has been to try and handle the pests themselves. Current systems include pesticides and chemicals, introducing natural enemies, and mechanical solutions. This combination of biological, cultural, mechanical, and chemical control is known as Integrated Pest Management or IPM **[15]**. Each of these comes with pros and cons. For example, the use of pesticides has been a controversial issue for a long time because of the potential dangers of using them. Pesticides are known to cause environmental harm, such as damage to soil and non-target species. As for mechanical solutions, they can cause issues because they are indiscriminate. A trap doesn't know if it's removing a potentially beneficial insect vs. a pests. The maintenance of traps can also become burdensome. The tradeoff between cost and effectiveness of mechanical methods can also be difficult to justify. Articles show that this type of pest control can cost up to $1200 per year, and that's with the scope being a home setting **[18]**. Farmers certainly have more space to worry about controlling **.** Finally, there is cultural pest control**.** The goal of this is to make the environment less suitable for pests. Preventative measures are used to stop pests before they even become a problem. The downsides of this are some methods have been found to cause erosion problems and be harmful for fish **[20]**. In addition, these methods are extremely specific, so almost the opposite of mechanical solutions, they may limit one pest while ineffective for a closely related pest.

While it's important to consider the large scale consequences of our design, we think it is better to start off with smaller applications. Using the principles of agile project management, we can start simple and gradually improve our product to meet varying needs. This method has been proven to reduce waste and improve total turnaround time of a project.

Since we will focus on smaller-scale applications, the home garden will be the first place our design can be used. Classical scarecrows are often the deterrence system of choice for gardens. While there is data to support the effectiveness of scarecrows, nearly every study also shows that they are a short term solution **[16],[17]**. Over time, birds and other pests have been shown to get used to the scarecrow's presence. We are proposing a more permanent and cleaner solution. The main deterrence in our design will be acoustic frequencies and not visible, this will keep the garden looking more minimal, as if nothing is there besides the plants. Another benefit of our system is that unlike scarecrows, it will be effective long-term.

In order to accomplish this design, we will break down the process into phases. The three phases shown below clearly identify how the autonomous acoustic deterrence system will be made.

System Design

Structural Drawings

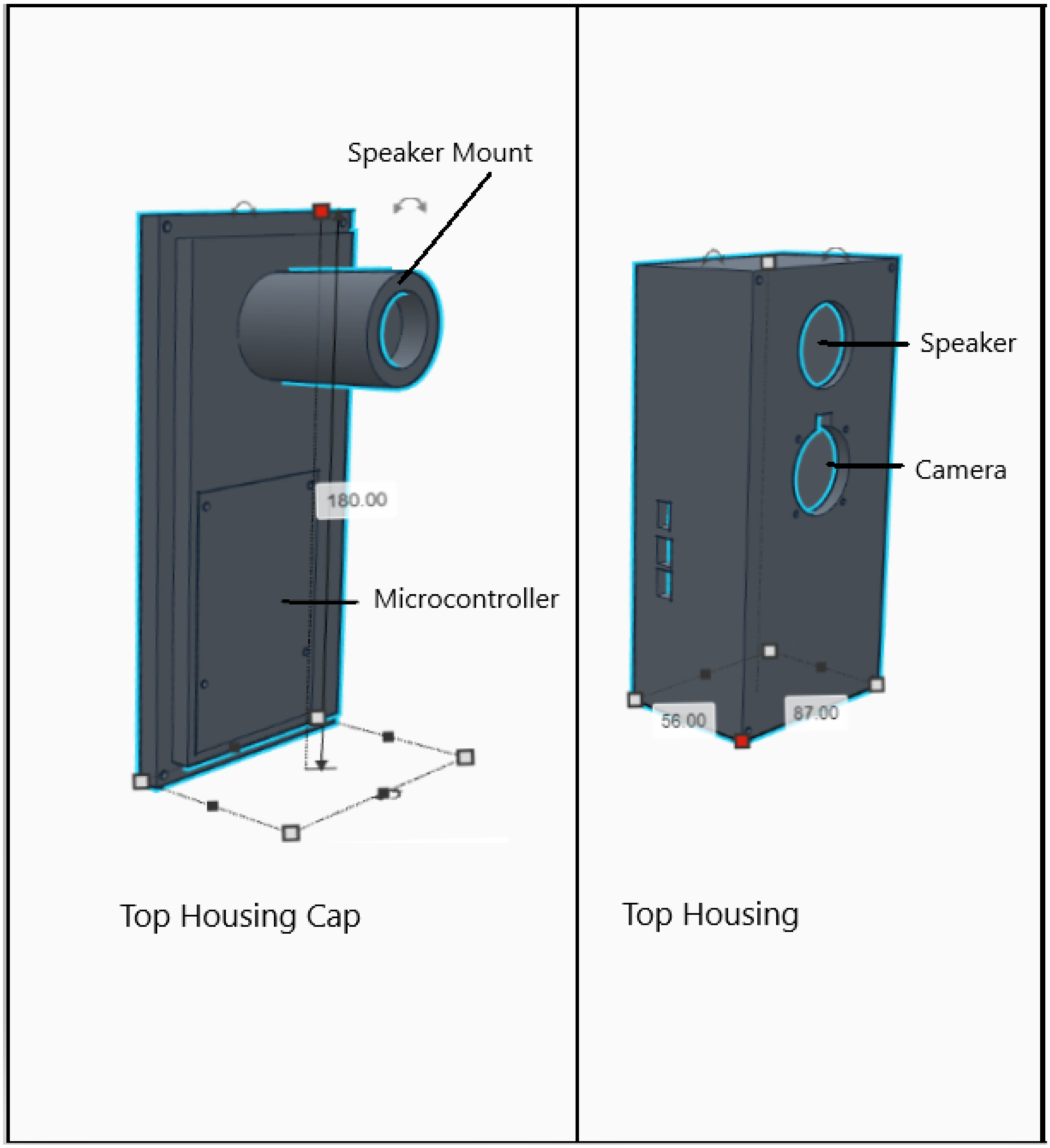


Figure 1. Top Housing: Contains Microcontroller, camera, speaker (all measurements in millimeters)

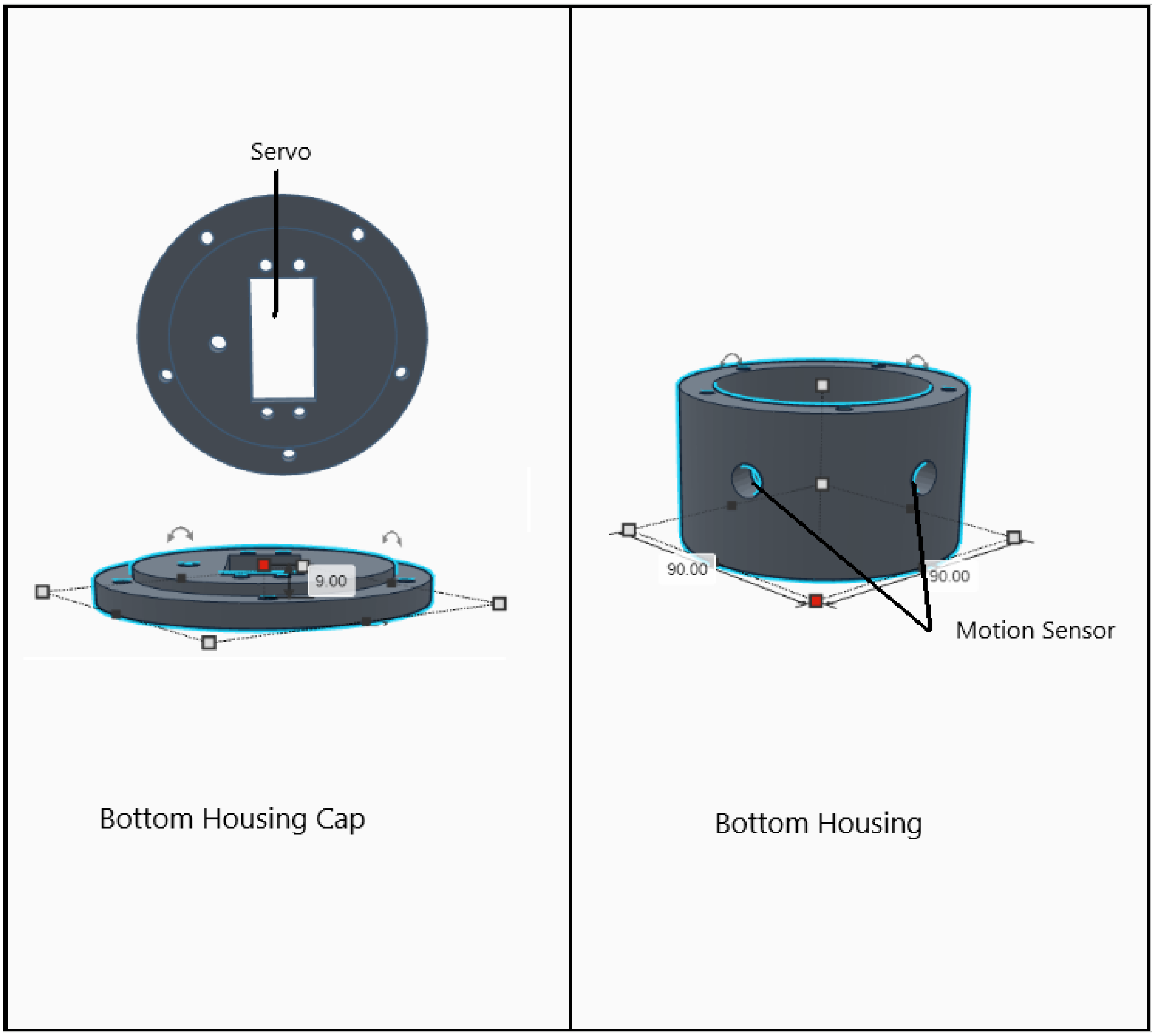


Figure 2. Bottom Housing: Contains Motion Sensors, Servo(all measurements in millimeters)

Final System Design

Figure 3. Physical implementation of System

Electrical Schematic

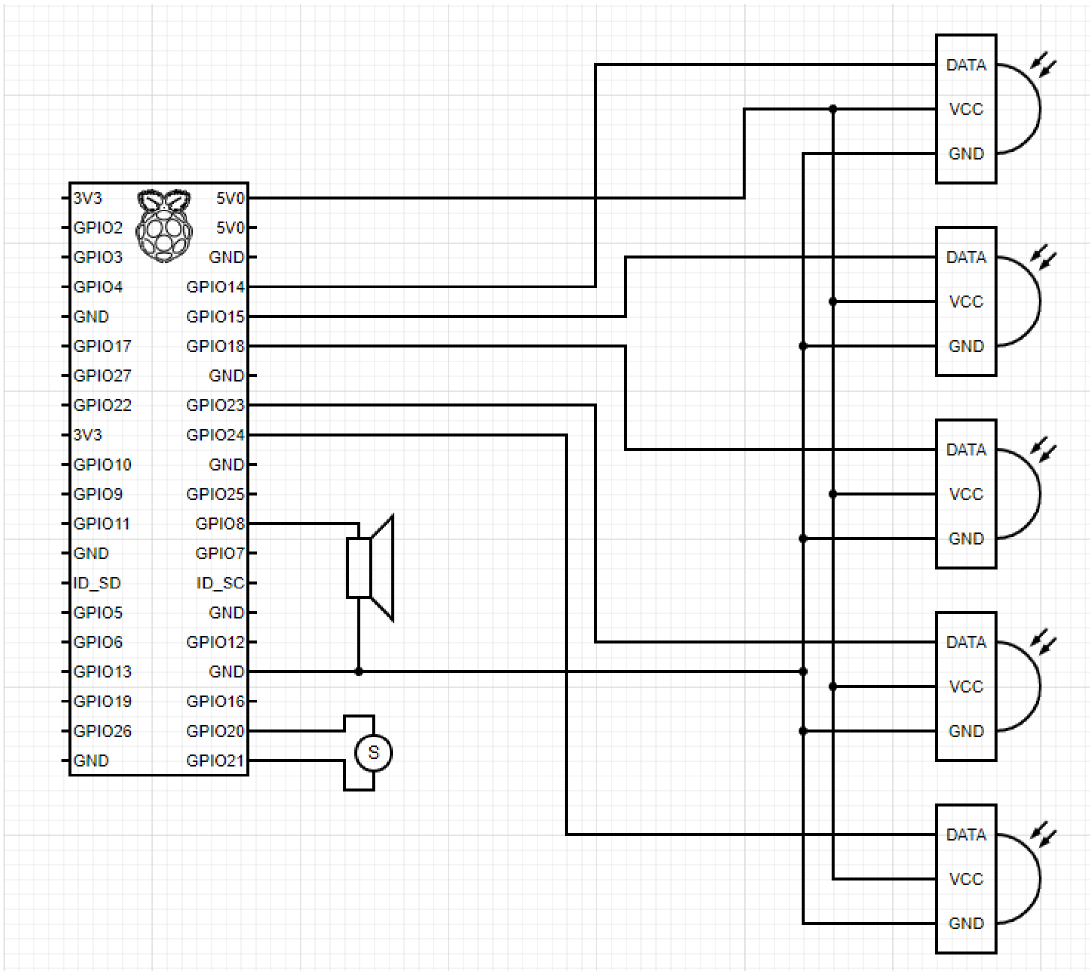


Figure 4. Electrical Schematic of system, includes Motion sensors, speaker, and servo connected to microcontroller

Components

Component	Brand	Data
Microcontroller	Raspberry Pi 4	https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf
Motion Sensor	EKMC1603111	https://www.mouser.com/datasheet/2/315/PANA_S_A00043955391-2560640.pdf
Camera	12.3 megapixel Sony IMX477 sensor	https://www.raspberrypi.com/documentation/accessories/camera.html
Lense	Adafruit 6mm Wide angle Lense	https://www.arducam.com/sony/imx477/
Speaker	Adafruit 3968	https://cdn-shop.adafruit.com/product-files/3968/Datasheet.pdf
Servo Motor	TD-8130MG	https://www.amazon.com/dp/B07Z3VGZNP?ref=ppx_yo2ov_dt_b_product_details&th=1

Figure 5. Component List

Power Consumption

No target detected (Stand-by Mode)= 4.27 Watts
Pi (idling)= 4.22 Watts
Motion Sensors = 0.05 Watts

Target Detected (Active)= 21.37 Watts
Pi (active) = 5 Watts = 5 volts at 1 amp
Motion Sensors = 0.05 Watts
Servo Motor = 16.32 Watts = 4.8 volts at 3.8 amps

Battery Life: 5000mAh at 5 volts will last 1 hour of constant active use, this includes constantly detecting a target, directing to target, image recognition, and speakers.

Methodology

Block Diagram

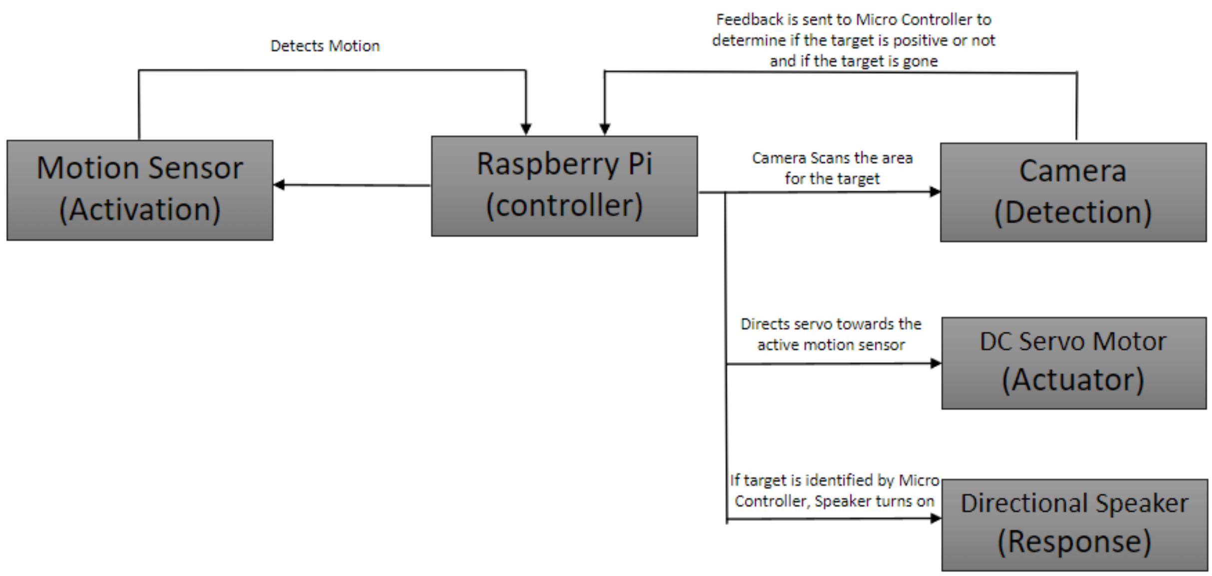


Figure 6. Block Diagram of the System

The system is housed within a centralized unit located at the center of the monitored area. This unit comprises a microcontroller that is surrounded by exterior motion sensors, while a servo equipped with a camera and speakers is affixed to the top. A Raspberry Pi 4 serves as the microcontroller and coordinates all system components. Upon sensing motion, the motion sensors transmit a signal to the microcontroller, activating the system. The servo then rotates to face the detected movement, orienting the camera and speakers accordingly. The camera scans the area for pre programmed targets, and if a match is detected, the microcontroller triggers the speaker to emit a specific frequency. After the speaker concludes its task, the camera reassesses the area for any residual targets. If the area remains positive for a target, the speaker is instructed to repeat the task; if not, the camera, speaker, and servo motor deactivate, and the system returns to sleep mode, solely scanning for movement.

Software Design

Flow Chart

This flowchart depicts the software code for the animal deterrence system. The program initiates by activating the camera to account for its slow start-up time from the Raspberry Pi. Subsequently, the program proceeds to verify if the object is within the range of the motion detector. If the object is within the range, the servo rotates towards the location of the animal. However, if the object is not within range, the program loops continuously, attempting to detect the object within the range of the motion detector. Once the unit is aligned with the animal's direction, the animal recognition algorithm is executed immediately to recognize the animal. We play a random sound of a specific frequency for a duration of 2 seconds if the animal detected is one of the animals in our dataset. If the animal detected is a bird, a random sound at 14kHz is played. If it is a squirrel, a sound at 22kHz is played, and if it is a raccoon, a sound at 30kHz is played. However, if it is not an animal in our dataset, we play a default sound at 10kHz.

The duration of playing sounds for more than 2 seconds or utilizing a long version of the same frequency is to prevent habituation. The system then checks if the animal has moved out of the way of the motion detector or moved out of the camera frame to determine if the deterrence system succeeded or failed. The recorded data, including timestamps and other relevant information, is stored in a spreadsheet. The program then loops back to checking if the object is within the range of the motion detector to repeat the process.

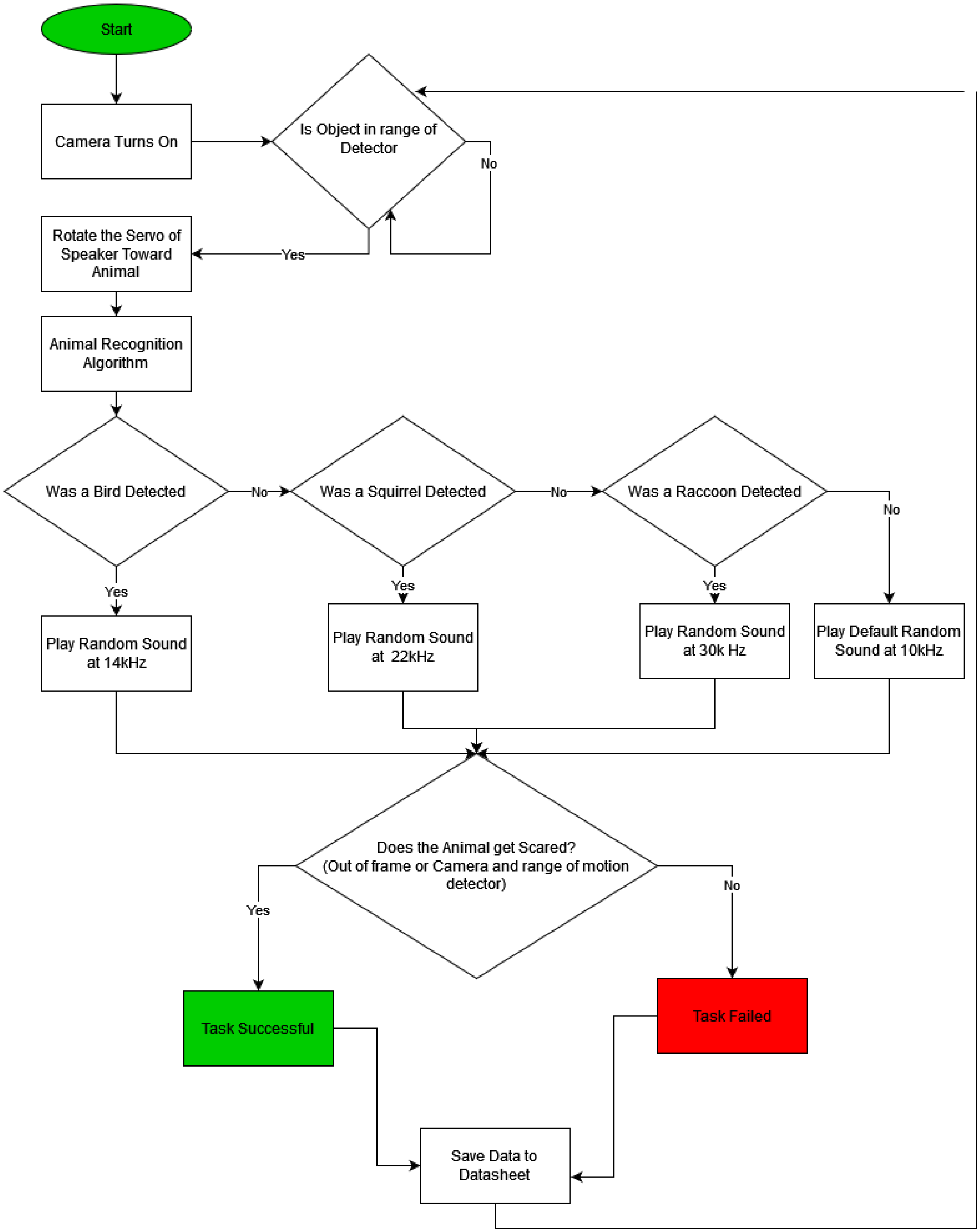


Figure 7. Flow Chart of the System

Computer Vision with Deep Learning

Background

Deep learning enables computational models with several processing layers to learn and represent data with various degrees of abstraction in a manner that is comparable to how the brain receives and comprehends multimodal information, implicitly capturing complex structures of big data. This is why deep learning is a branch of artificial intelligence or the attempt to mimic intelligence using a machine. The definition here for intelligence being the ability to retain and remember information. Neural networks, hierarchical probabilistic models, and other unsupervised and supervised feature learning algorithms are all part of the broad family of techniques known as deep learning. Deep learning approaches have seen a recent uptick in interest due to their demonstrated superior performance than previous state-of-the-art methods in a number of tasks as well as the amount of complicated data from various sources (e.g., visual, audio, medical, social, and sensor). Many computer vision issues, including object detection, motion tracking, action recognition, human position estimation, and semantic segmentation, have greatly benefited from deep learning.

The two key learning paradigms in image processing tasks are supervised and unsupervised learning. Supervised learning is learning through pre-labelled inputs, which act as targets. For each training example there will be a set of input values (vectors) and one or more associated designated output values. The goal of this form of training is to reduce the models overall classification error, through correct calculation of the output value of training example by

Training. Unsupervised learning differs in that the training set does not include any labels. Success is usually determined by whether the network is able to reduce or increase an associated cost function. However, it is important to note that most image-focused pattern-recognition tasks usually depend on classification using supervised learning.

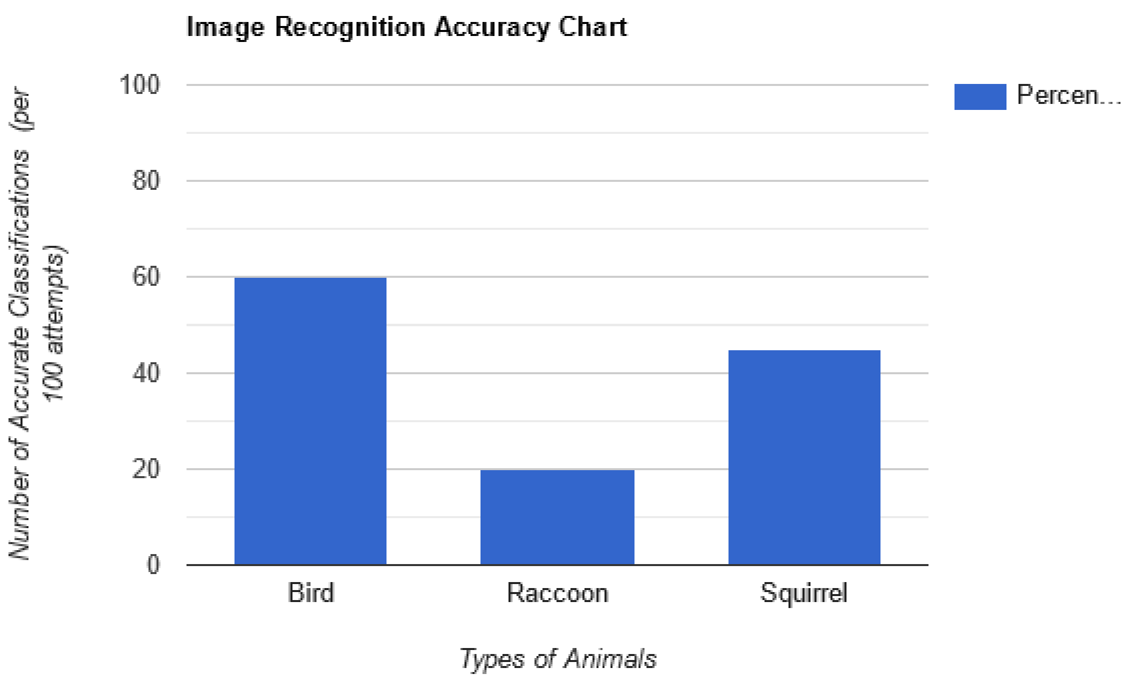
Implementation

We will be utilizing the Single Shot Detector (SSD) algorithm, which is a variant of convolutional neural networks (CNNs), for computer vision tasks on a Raspberry Pi 4. Using TensorFlow as the implementation framework will allow us to achieve our goals since previous projects have demonstrated its compatibility with the Raspberry Pi 4. CNNs are ideal for image processing since they have the ability to automatically learn features, rather than relying on manual features like other network types. Additionally, CNNs are invariant to transformations like translation, scaling, and rotation, which makes them particularly well-suited for computer vision problems like object detection. The SSD algorithm, in particular, uses a combination of convolutional layers and pooling layers to predict the presence and location of objects in an image in real-time, making it suitable for our needs. However, CNNs require labeled training data for optimal performance, whereas other network types like DBNs/DBMs and SAs can operate without labeled data. For our project, we trained the system to recognize pests such as birds, raccoons, and squirrels using images of these animals as input. CNN's ability to be invariant to changes in the image, such as scaling, is crucial since we were dealing with live animals of different sizes that move around.

Testing Data / Results

Accuracy of Image Recognition Algorithm

To assess the accuracy of the image recognition algorithm, the camera and Raspberry Pi components were isolated from the rest of the system. To ensure reliable results, specific videos/livestreams of each animal were used for testing, with the image recognition software active for 100 attempts per animal. The following data presents the number of accurately classified animals out of 100 attempts by the image recognition algorithm.



Based on the given data, the image recognition software was most accurate in identifying birds, with an accuracy rate of 60% out of 100 attempts. Squirrels had a lower accuracy rate of 45%, while raccoons had the lowest accuracy rate of 20%.

This data suggests that the image recognition software was better at identifying birds than it was at identifying squirrels and raccoons. However, the accuracy rate for all three animals was relatively low, indicating that there is room for improvement in the image recognition algorithm.

Several factors may have contributed to these accuracy rates, including the quality of the input images, the complexity of the image recognition algorithm, and the size of the dataset used to train the algorithm. Improving any of these factors could potentially improve the accuracy of the image recognition software.

It is also worth noting that this data was collected over 100 attempts for each animal, and it is possible that the accuracy rates could vary if more attempts were made. Additionally, the graph could be more informative if it included error bars or other measures of variability, which would provide more information about the precision of the accuracy rates.

Conclusion

Limitations

Some limitations of our system include the range of the camera, response time of the actuators, and the overall effectiveness. The first two limitations play heavily into the third. When it comes to the camera range, it would be better if animals were able to be detected from further distances in order to stop them from entering a predetermined area around the garden. The way our system works now, it isn't until the animal is within close range that the camera can spot it. This can be improved by more strategic placement of the system, however, the best solution is an improved camera with better range. Or even multiple cameras.

The response time of the motion sensor as well as the servo motor and buzzer was a major limitation of the system. While doing testing, we observed how fast backyard pests really are. For example, birds were able to fly in, take some crops and fly away, all within a few seconds. This didn't give all the moving parts of our system time to work effectively. The detection, and especially the servo take a long time respond, resulting in poor results at times

Future Enhancements

There are many improvements that could be made to the system. The primary ones include better components, dynamic deterrence, and scalability.

Better components means a more powerful servo, better ranged camera, and more effective speaker. Due to budget constraints in this project, we had to settle for parts that were workable at best. With improved specs on our components, the code would interact much faster and more efficiently with the actuators, resulting in more effectiveness.

Dynamic deterrence would be the ability of our buzzer frequencies to change periodically. This would help the system be effective in the long term. In our initial research, we found that animals can become immune to the effects of frequencies. So by changing the frequency dynamically, our system would overcome this challenge.

Scalability means being able to take our systems applications to a larger area. We tested in the backyard, but with improvements we hope this system can be used for farming and agricultural applications. That is where the real power of this technology comes into play. Of course, backyard gardening is important, however, most people who have the luxury of using AI in their backyard garden aren't in dire need of food. But by taking our system to farms, we can impact people who are negatively affected by pests.

Learning Outcomes

The biggest lessons we learned from this project were leadership, the value of time management, and the importance of testing.

When it comes to leadership, we saw how having a person who was on top of things really improved the workflow of the project. Without a designated leader, it is very difficult for a project to run smoothly. Our leader was able to keep us on tasks and was also critical in the development of the system. Although there was only one assigned team leader, all members of our team learned lessons about how to take command and get a project done.

Time management was the most important thing we learned. In addition to working on this project, we all were finishing the spring semester in school. This meant studying for exams, and even finishing other capstone projects. We quickly saw the value of weekly meetings with our team, as well as getting updates from our instructor. There was a time in the middle of the semester that we slowed down, but we were able to recover towards the end. The biggest takeaway in terms of time management is this, always be working on something. There were many times when one part of the project relied upon another part of the project's completion. Rather than simply waiting around, we learned that the other team members should be actively involved, as this saves stress later on.

Finally, the importance of testing was clear toward the end of the project. For a long time, we had an idea of how this system would work. It wasn't until we tested that we saw many flaws that required solutions. For example, we didn't anticipate the servo motor being the biggest issue in our build process. We learned that waiting until the very end to test isn't a good plan. Rather, the system should be built using abstraction and modularity. So each part should build on a previous part, and testing should be done at all phases. This gives confidence that the overall system will work when put together one piece at a time rather than all at once.

Bibliography

1. E. Zahir, M. A. Hossen, M. A. Al Mamun, Y. M. Amin and S. M. Ishfaq, "Implementation and performance comparison for two versions of eye tracking based robotic arm movement," 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE), 2017, pp. 203-208, doi: 10.1109/ECACE.2017.7912906.
2. R. R. Mussabayev, "Colour-based object detection, inverse kinematics algorithms and pinhole camera model for controlling robotic arm movement system," 2015 Twelve International Conference on Electronics Computer and Computation (ICECCO), 2015, pp. 1-9, doi: 10.1109/ICECCO.2015.7416879.
3. R. Szabó and A. Gontean, "Robotic arm autonomous movement in 3D space using stereo image recognition in Linux," 2014 11th International Symposium on Electronics and Telecommunications (ISETC), 2014, pp. 1-4, doi: 10.1109/ISETC.2014.7010792.
4. *How Well Do Dogs and Other Animals Hear?* Frequency hearing ranges in dogs and other species. (n.d.). Retrieved October 5, 2022, from https://www.lsu.edu/deafness/HearingRange.html

5. “Effects of Noise on Wildlife and Other Animals: Review of Research Since 1971,” *EPA*. [Online]. Available: https://nepis.epa.gov/Exe/ZyPURL.cgi?Dockey=9100ONX4.TXT. [Accessed: 05-Oct-2022].
6. A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, “Deep Learning for Computer Vision: A brief review,” *Computational Intelligence and Neuroscience*, 01-Feb-2018. [Online]. Available: https://www.hindawi.com/journals/cin/2018/7068349/. [Accessed: 21-Nov-2022].
7. W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bulletin of Mathematical Biology*, vol. 5, no. 4, pp. 115–133, 1943.
8. Y. LeCun, B. Boser, J. Denker et al., “Handwritten digit recognition with a back-propagation network,” in *Advances in Neural Information Processing Systems 2 (NIPS*89)*, D. Touretzky, Ed., Denver, CO, USA, 1990.
9. S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
10. G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
11. “Raspberry Pi and computers-based face detection and recognition system,” *IEEE Xplore*. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8398677 . [Accessed: 22-Nov-2022].
12. M. Sajjad *et al.*, “Raspberry Pi assisted face recognition framework for enhanced law-enforcement services in smart cities,” *Future Generation Computer Systems*, Nov. 2017, doi: 10.1016/j.future.2017.11.013.
13. Thomas harrisson, “Rise in insect pests under climate change to hit crop yields, study says,” *Carbon Brief*, Aug. 30, 2018. https://www.carbonbrief.org/rise-in-insect-pests-under-climate-change-to-hit-crop-yields-

study-says/#:~:text=At%20present%2C%20around%2010%2D16 (accessed Nov. 22, 2022).

14. FAO - News Article: Climate change fans spread of pests and threatens plants and crops, new FAO study," *www.fao.org*. https://www.fao.org/news/story/en/item/1402920/icode/
15. What Is Integrated Pest Management (IPM)? / UC Statewide IPM Program (UC IPM)," *www2.ipm.ucanr.edu*. https://www2.ipm.ucanr.edu/What-is-IPM/#:~:text=IPM%20is%20an%20ecosystem%2Dbased
16. C. Sweeney, "10 Scarecrows for the 21st-Century Farmer," *Popular Mechanics*, Oct. 18, 2009. https://www.popularmechanics.com/home/lawn-garden/how-to/a4682/4334347/ (accessed May 09, 2019).
17. Do Scarecrows Actually Work?," *Mentalfloss.com*, Oct. 29, 2015. https://www.mentalfloss.com/article/70232/do-scarecrows-actually-work
18. How Much Does Pest Control Cost? (2022)," *This Old House*. https://www.thisoldhouse.com/pest-control/reviews/pest-control-cost#:~:text=If%20you%20only%20receive%20one (accessed Nov. 22, 2022).
19. Cultural Control | Wisconsin Vegetable Entomology," *vegento.russell.wisc.edu*. https://vegento.russell.wisc.edu/ipm/cultural-control/#:~:text=The%20goal%20of%20cultural%20control (accessed Nov. 22, 2022).
20. Cultural methods of pest, primarily unsect, control," *Mcgill.ca*, 2020. https://eap.mcgill.ca/publications/eap58.htm
21. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for Mobile Vision Applications," *arXiv.org*, 17-Apr-2017. [Online]. Available: https://arxiv.org/abs/1704.04861# . [Accessed: 22-Nov-2022].

22. A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, “Deep Learning for Computer Vision: A brief review,” *Computational Intelligence and Neuroscience*, 01-Feb-2018. [Online]. Available: https://www.hindawi.com/journals/cin/2018/7068349 /. [Accessed: 21-Nov-2022].
23. A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, “Deep Learning for Computer Vision: A brief review,” *Computational Intelligence and Neuroscience*, 01-Feb-2018. [Online]. Available: https://www.hindawi.com/journals/cin/2018/7068349/ . [Accessed: 22-Nov-2022].
24. “How to perform object detection with tensorflow lite on raspberry pi,” *Digi*. [Online]. Available: https://www.digikey.com/en/maker/projects/how-to-perform-object-detection-with-tensorflow-lite-on-raspberry-pi/b929e1519c7c43d5b2c6f89984883588 . [Accessed: 22-Nov-2022].
25. “Object detection,” *Wikipedia*, 23-Sep-2022. [Online]. Available: https://en.wikipedia.org/wiki/Object_detection. [Accessed: 22-Nov-2022].
26. T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, “Microsoft Coco: Common Objects in Context,” *arXiv.org*, 21-Feb-2015. [Online]. Available: https://arxiv.org/abs/1405.0312 . [Accessed: 22-Nov-2022].
27. “Transfer learning,” *Wikipedia*, 28-Sep-2022. [Online]. Available: https://en.wikipedia.org/wiki/Transfer_learning#:~:text=Transfer%20learning%20(TL)%20is%20a,when%20trying%20to%20recognize%20trucks . [Accessed: 22-Nov-2022].
28. “Gardening Statistics in 2022 (Incl.. Covid & Millennials).” *Garden Pals*, 9 May 2022, https://gardenpals.com/gardening-statistics/