Sectioned Crop Management System

Project Group Unknown

Veselin Daskalov

Christos Constantinou

Nikolay Pavlov

Alexis Philippou

Alexandra Pintilie

Nikola Marinov Yordanov

February 1, 2020

1 Problem: Efficient Crop Management

The problem we identified is efficient crop management. Said as such the problem seems to vague and it is, however we have identified key aspects of this problem that we will try to tackle.

The first problem is efficient water distribution. This is quite a very widespread problem, especially in very dry places. Today water is distributed between farmers equally. While this is good in theory in practice this could lead to trouble. This is because some soil is naturally dryer and some crops require more water than others. However with the current system all crops and all soil receive and equal amount of water, this both wastes water and incurs huge costs to the farmers.

The next aspect is pest control. Due to the foraging behavior of birds, rodents and other such pests, crop damage is quite common. What has not been known however is how much revenue has been lost from this. A recent study found that farmers can lose up to 16% of their crops. This both incurs massive costs to farmers and forces them to buy expensive contraptions to try and deal with this problem, contraptions which might not work in the end. This problem is also not limited to just one part of the world, thus any solution will need to be deployable anywhere it is needed.

2 Solution: Section Based Crop Management System

The solution we decided to implement is the Section Based Crop Management System. This system will focus on repelling pests such as birds and rodents. We chose to focus on the pest problem for two reasons. The problem is far more widespread while also being far less understood and the systems currently in place to deal with the problem are either expensive or ineffective

Our solution is to create a system that is able to reliably detect pests and respond with sound in order to repel them. We plan on making this system truly a smart environment by networking all of the different nodes together and from there be able to collect statistical data to identify problem areas where more attention needs to be diverted.

The detection will be be done via a object detection algorithm. We will utilize a camera and a Computer Vision algorithm. This algorithm should be able to differentiate between pests and other objects.

The networking part should be able to handle two-way communication as well as nodes being able to access and leave the network as will. This should also extend to the section level, with any section being able to be shutdown for maintenance or redrawing as needed without the whole network needing to be shutdown.

Lastly the physical setup should not require much power and be as portable as it could be, without compromising on any other aspects of the system.

3 Method and Tools Used

The system we plan would require quite a lot of processing power as well as support for high level languages. This is why we are going to use the Raspberry Pi 3 B+, for the camera we will use the Pi Camera v2.1. The Raspberry Pi 3 B+ has on-board Bluetooth and Wi-Fi with functionality of being setup as and Access Point. As for the Pi Camera it has out of the box support on the Pi as well as libraries we could use in our language of choice which is Python3.

Due to the complexity of the proposed system we have split our implementation into a basic and advanced approach.

3.1 Basic Approach

The basic approach will mostly focus on implementing the network and object detection algorithm with the least amount of complexity, while still being able to perform their functions effectively.

The basic networking setup would require a server through which all nodes in one area will be connected to, this will be referred to as a section. The server will not be capable of detecting pests and while being the only node that will be able to communicate to any other sections. The server will also store the statistics of the section. The clients will be the detecting nodes, they will only be connected to the server and will have to pass any messages through the server if they want to reach other nodes. The connection will be setup via WiFi when a client connects to a server and via Ethernet for server to server, or section to section. We will use WiFi because it is the most convenient for this application, however we will use Ethernet for server to server because it has better range and reliability.

The detection algorithm will be a combination between a motion detection algorithm and a second algorithm that acts upon the objects in motion. The motion detection algorithm will be the most complex part of the project so we are looking into the possibility of finding an already made one. As for the object recognition we are still debating on how to go about it, to start off with we plan trying object filtering based on the dimensions of the detected object. If that doesn't give us the needed accuracy rate then we will explore other possibilities. Lastly on the subject of AI we plan on using OpenCV along with this algorithm and apply the object filtering on top of it.

3.2 Advanced Approach

The advanced approach will concern itself with outlining the setup that will result in the most optimal system for the given task. It will be the easiest to convert into an actual product and all parts of it will be custom made. This will, however, also result in much higher complexity which might render the project unfinishable by the deadline.

The networking will probably have the biggest change in terms of how it works. In the advanced approach there will be no server nodes. All of the network will be comprised of detection nodes that will be networked in a peer-to-peer network. The network will be able to handle if nodes go down unexpectedly or any new nodes are added, without needing to restart. We still want sectioning to be possible so all groups of nodes that close together will be grouped into a section. This will be marked within the setup process. Statistics gathering will be done by each individual node and will be accessible anywhere in the network.

The detection algorithm will not be that different in function, however it will be far more complex. For the advanced detection algorithm we plan on combining the motion detection and object recognition algorithms into one algorithm that uses machine learning to optimize its detection rate. This could be aided by libraries such as TensorFlow which also has a version for the Pi, however the computational requirements coupled with the networking stress the Pi more than expected. This is why we plan on developing the algorithm with the aid of TensorFlow Light, a fork of TensorFlow optimize for the Pi. This will also help us greatly in the development and training of the algorithm.

This is why we are also adding to this advanced approach a hardware requirement to the enclosure that being active cooling. This will allow us to overclock the Pi possibly giving us the performance we need to run the algorithm and the networking concurrently.

4 Results and Conclusions

The project consists of four main aspects, which are: physical setup, networking, bird repellent and bird detection (Computer Vision). The results for each are as follows:

4.1 Physical Setup

For our physical setup we used two Raspberry Pis, with one in set in AP (Access Point) mode and the other connecting to the client. This setup worked flawlessly and the Pis had enough processing power to handle both the motion detection algorithm and the networking aspect. There were no real problems with interference from other signals with one exception regarding the range we got that was roughly 170 meters. This means that at minimum there are 100 meters of signal which can sustain two way communication.

4.2 Networking

The networking had few problems as well. Due to the way we had setup the multi client support every client is handled in a separate thread, thus there is a limit to how many threads you can make on a computer such as Pi. In our testing at around 10 threads we noticed some slight performance issues so we capped it off at that. Moreover, we tested if the server could double up as another detection node, yet the Pi started to thermal throttle at that point. This could be fixed by better cooling, however for our purposes the difference would be negligible.

4.3 Bird Repellent

The bird repellent system uses both ultrasonic and predator sounds. The sound files are stored in pointer which stores them within an array. We have done this because while testing it we realized that it is quite difficult to change the sound files when a new one became available or one sound file is not needed any more. As for the efficiency of the system we unfortunately did not have much time to test this. In spite of this, we manage to find that the predator and scare sounds (sounds designed to scare birds) had much more effect as compared to the ultrasonic sounds. Aside from that the system does not have any major issues which need to be addressed.

4.4 Bird Detection

For this we used a motion detection algorithm which we found online. We then loop over all the detected objects and filtered them to only have the birds left. During testing the algorithm performed well, but it is a background detection algorithm so any objects that move too close will cause the algorithm to act erratic. As for the performance, we have tried to use packages for numerical optimization as well as limiting the framerate of the algorithm. With all that said we still sustain 60% CPU load giving us little room to do much else. Lastly, for this algorithm to work as intended the first frame of the video must be empty, this will be the reference frame. This is important to state, because if not properly setup the algorithm will not work at all. With all this said, here are some examples of the algorithm; keep in mind that the only options for the algorithm are a person being detected or a bird.

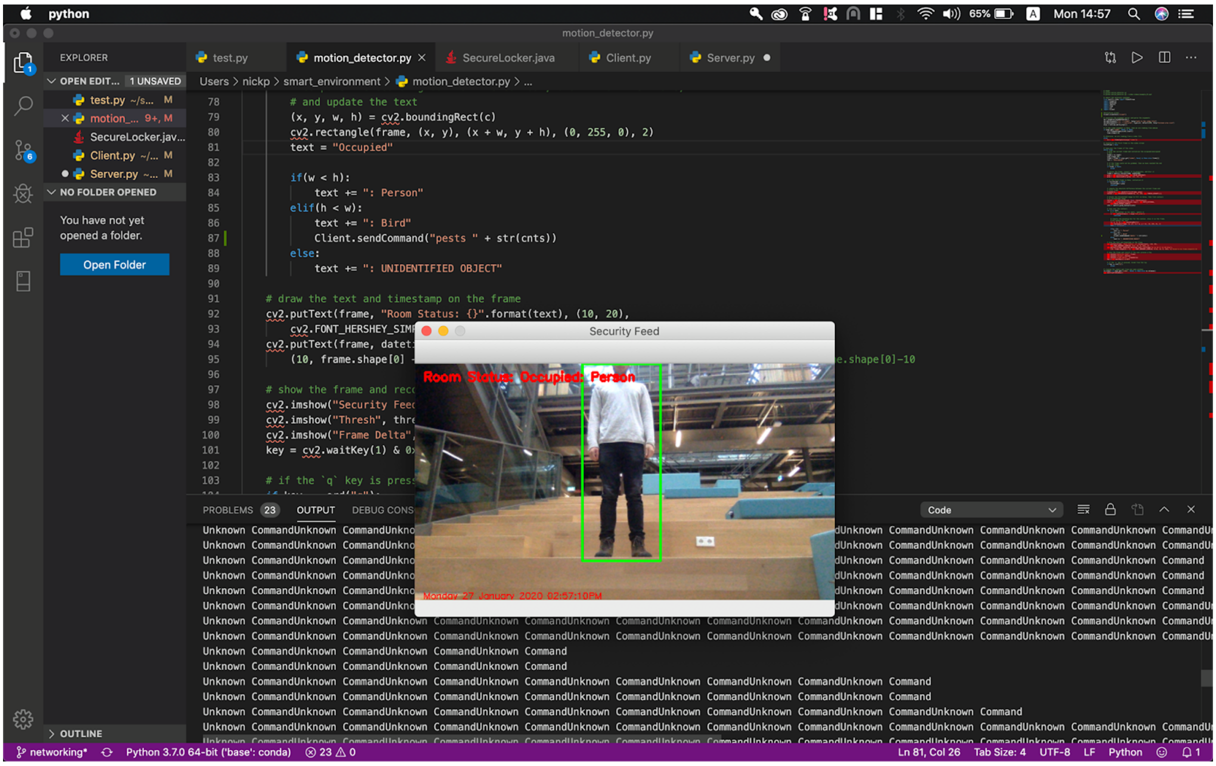

Figure 1: System Detects a Person

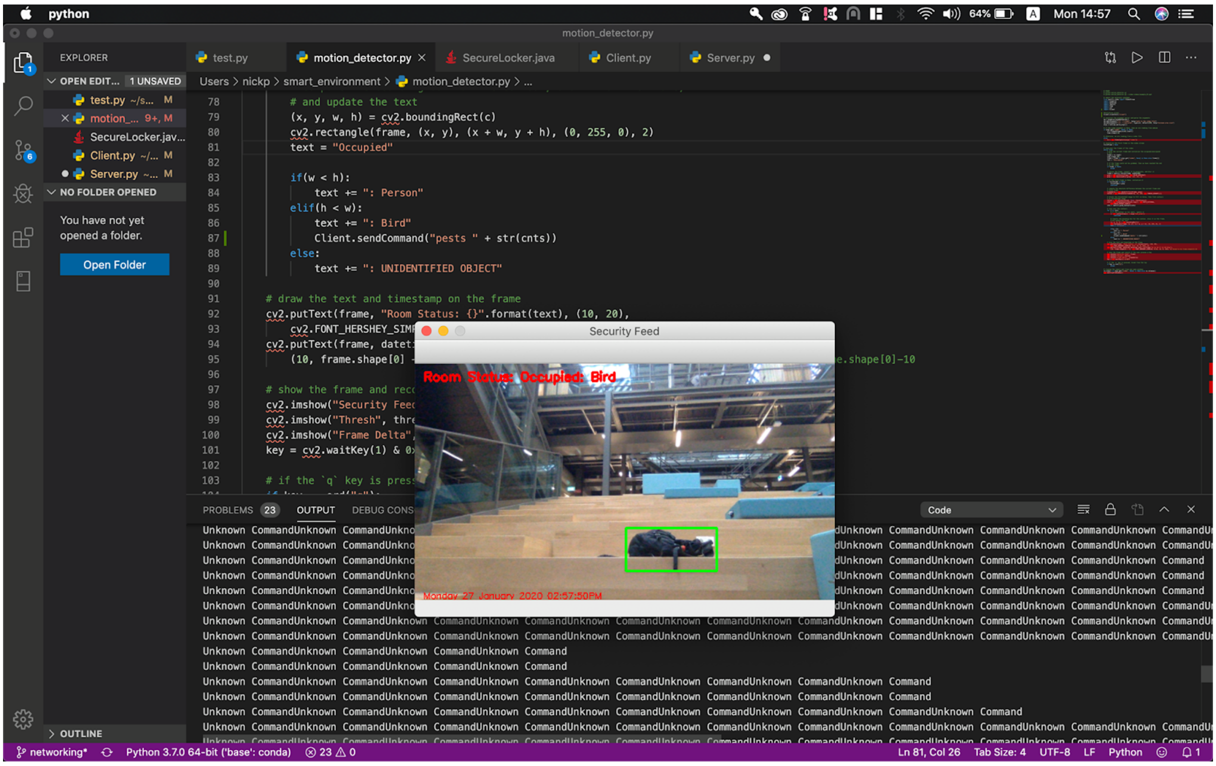

Figure 2: System Detects Bird Example 1

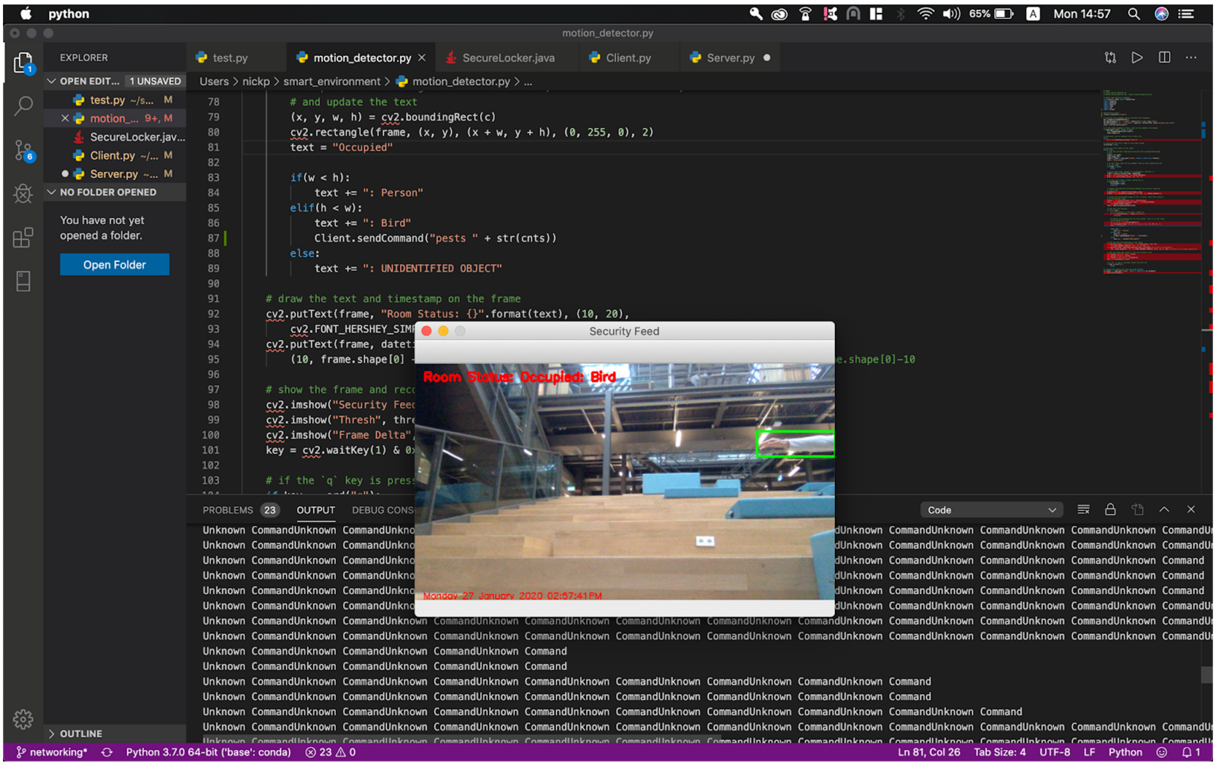

Figure 3: System Detects Bird Example 2

4.5 Conclusion and Final Thoughts

After doing all the testing and development we are happy with how the project turned out. Looking back, we realise that dividing the project into a basic and advanced approach really helped us out by giving us clear organization and perspective. Now the system does have some flaws, mainly the very precarious nature of the setup but aside from that. In retrospect, there are some things we would change mainly when started full development. There were many things that stopped us from fully committing, yet that still does not change the fact that more time would have helped with polish and bug testing.

As for the features we were able to add. As for now the system is able to detect birds using the camera and motion detector algorithm, then after looking at the dimensions of the detected object we can identify it. If the object is a pest the detecting node can send the information to a server which then transmits to all nodes connected to it to start emitting sound. The sound will either be ultrasonic noise or predator/loud sounds to scare off the pest, the sounds will also be randomized so that the pests don't get used to them. The server will act as a section and will be configured as an access point with which all detecting nodes connect to. The server is also able to store detection rates of different cameras and return that data to any requesting node. Then the server can be connected to other servers, or sections, for a whole network to be built, at this point only the node connected to the server can access the statistics of that section.

Lastly we want to touch on the commercial feasibility of the system. Overall what we have built needs very little modification to work as a product, the main feature it needs is a custom algorithm for detecting movement. If this is done then you have a fully working system for detecting birds. With some adjustments, depending on your object detection algorithm, adapting this to work with rodents would require simple adjustments to the algorithm and of course the thermal imaging equipment. Last part we need to make this a functional product is a housing that is weather resistant this is very important given the intended use case for this is farmland without any limitations on which part of the world it is just that the crop is supported.

5 Physical Setup

Here are some pictures of the final physical setup used and setup.

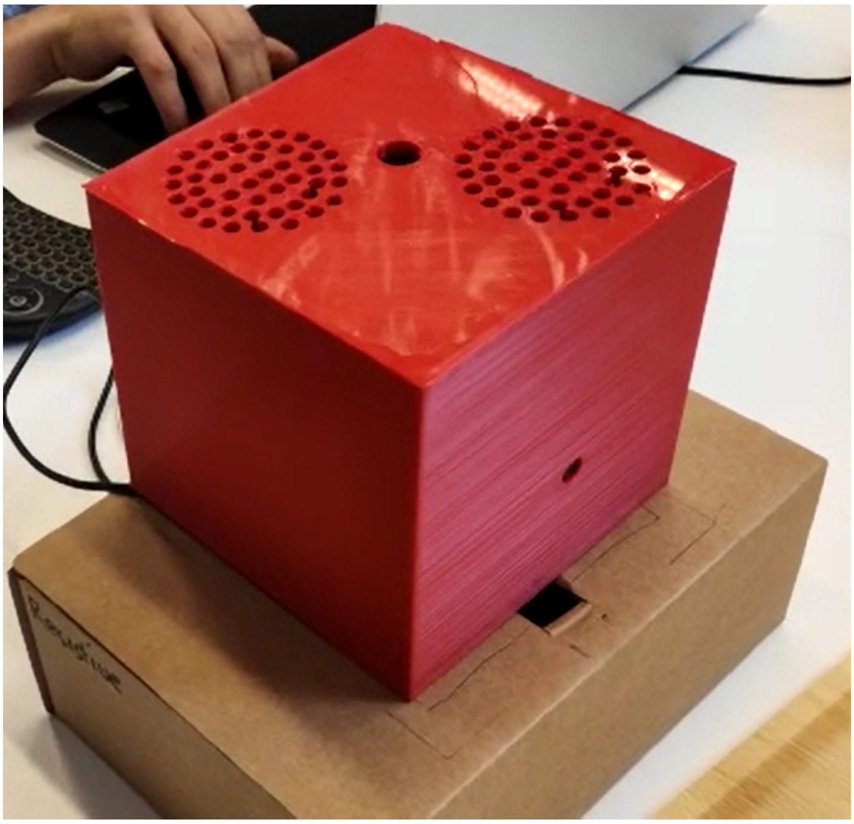

Figure 4: Detection Node Setup

Figure 5: Server Setup

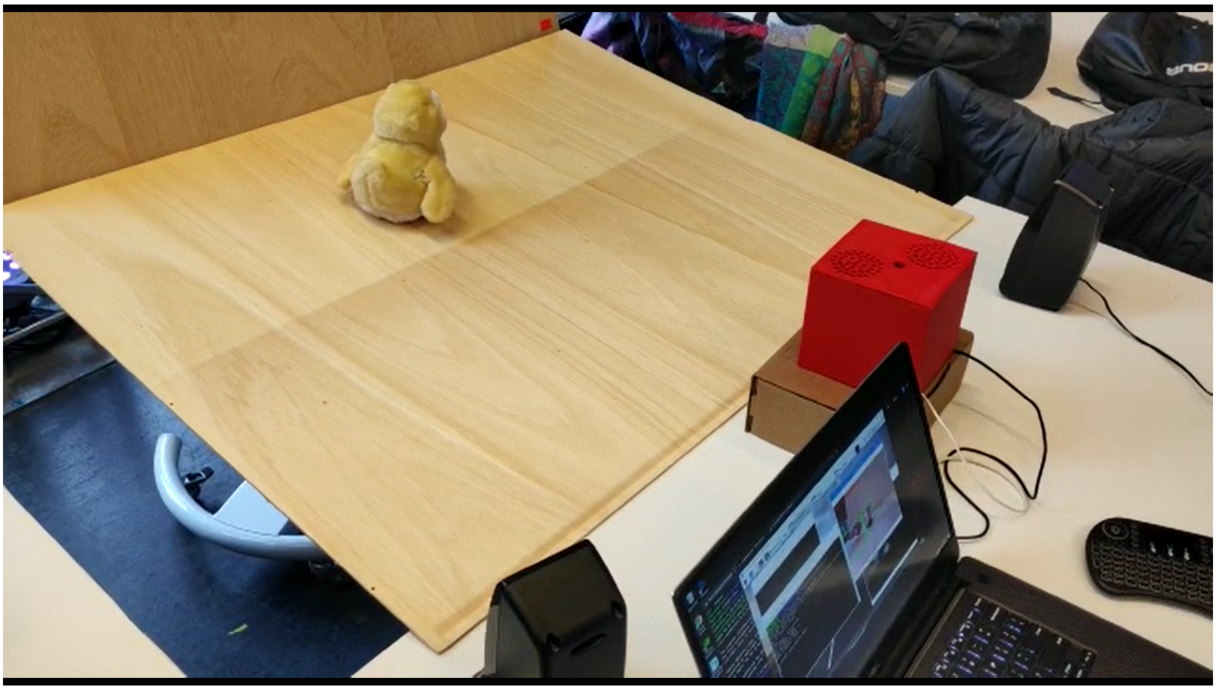

Figure 6: Full Setup