Project: Garden Guardians
Title: Garden Guardians
Group Number: 4
Team Members: Ching Zheng, Nimish Kapoor, Xiang Chen, Henry Nguyen, Jackson Lim

Abstract

The Garden Guardian is a project that will detect and identify mammalians pests and mitigate their presence in a home garden. When you see or hear the word "pest", you probably envision insects; however, the more threatening pests to a garden may be larger animals such as: rabbits, deers, raccoons, or potentially even your pets. Thus, for our project the main idea is to use live video feed to identify whether an animal is a pest, and if the application identifies an organism as a pest, our application will take defensive measures against it. Of course, there are alternatives compared to our solution, alternative solutions could include spraying a garden with chemicals and erecting a protective structure around the perimeter. However, these solutions are either short-termed or labor-intensive/expensive. So we believe our project to be a compromise, where the device and application are built for long-term use and the expenses are kept to a minimum. We hope that by applying our knowledge of machine learning, computer vision, and software development, we will be able to effectively protect everyone's produce.

Introduction

This project emerged from seeking ideas to apply our knowledge of machine learning and computer vision. Although many modern applications of computer vision such as self-driving cars and surveillance analysis sound exciting, many of the projects demanded frontier developers and researchers in the industry with long term planning and working with bountiful amounts of resources to invest in. To find a project that well suited our scale, we had to look for a more niche application: gardening. As with many other computer vision projects, this too has a facet in automation - with a goal to protect crops from animals. Relevant to many gardeners at home, and potentially new gardeners who feel the need to start after the quarantine due to the recent coronavirus, the product may help preserve other's efforts by saving their crops. The primary goal of this project is applying our knowledge and seeing it come to life in a familiar environment - and more importantly, to notice how computer vision can help our everyday life.

Related Work

- Patent for motion sensor lighting https://patents.google.com/patent/US5258899A/en
  - We have all seen these types of lights. Many of us may even have them but these lights work through rader and do not care for the subject and aren't likely to scare any pests. Our project on the other hand will only affect pests and is much more likely to produce results.

- Project using object detection and a deep learning network https://www.pyimagesearch.com/2017/09/11/object-detection-with-deep-learning-and-opencv/
    - Our project will work with a web server to lessen the stress put on the end users hardware.
- Project contains a CNN that classifies the CIFAR-10 image set.
    - https://towardsdatascience.com/image-classifier-cats-vs-dogs-with-convolutional-neural-networks-cnns-and-google-colabs-4e9af21ae7a8?gi=179edc908dac
    - We used this as reference when implementing our own CNN.
- A project that we want to use as comparison for our end results
    - https://towardsdatascience.com/detecting-animals-in-the-backyard-practical-application-of-deep-learning-c030d3263ba8
- A RESTful API project
    - https://www.guru99.com/restful-web-services.html
    - We will be using the same technology, but with NodeJS and our own features

Methodology

For the project we decided to use Jupyter Notebook as our main developmental tool. The reason why we decided to use Jupyter is primarily because for the first step, we plan to implement a convolutional neural network to classify images. We choose the CIFAR-10 image dataset as our starting dataset because it contains three species of animals that we want to identify: cats, dogs, and deers. Thus, it's relevant to the dataset we will be creating and classifying next. Once the CNN is trained, we will upload our own test images to test the accuracy of our CNN model. For the dataset, we will extract images we need from different available datasets found online.

If the resulting accuracy is not satisfiable, we will explore different hyperparameter tuning methods to improve the result. Possible ways are like modifying the layers (number of hidden layers, activation function, dropout, etc), changing the optimizer and loss function, and image preprocessing. Last but not least, we have also considered using a pre-trained CNN as the base model, instead of building our own from scratch. We can utilize an already trained model in a related problem on our dataset. Oftentimes in ML, one will need a lot of data, computational power, and time to build a deep network that can provide the best result. Due to the time limit, we don't have these resources. Thus, using a pre-trained model can speed up the learning and also provides reasonable results. The objective of the CNN is to identify and classify images of animals with a high accuracy rate. If the accuracy satisfies our standards, we will proceed to the second step of our project: object detection.

The purpose of the object detection is to detect the motion from a live video feed. The motion tracker works by using the first frame of the video and formatting it to be used as a template to be used as a comparison for successive frames. Essentially, what the program does is remove any details from a frame and convert it to a plain black and white image, where objects of interest are completely shaded in white and anything else is shaded black. Once black and white image is obtained it can then be used to determine motion. Then we find the contours in the threshold frame and if the area of the contour is above a predefined level we can consider the object a large enough change to determine motion. Finally we draw a box around the object to track its movements.

Once the two parts of the project are individually complete, we plan to merge the two aspects of the project together into a single application. The plan is to use the object detection function to take the first frame of a video feed with an animal and upload it to our CNN to be processed and identified. If the CNN identifies the animal as an animal that falls under the pest category the protective measures will activate. Since we do not plan to implement the hardware aspect of this project, a web API will take its place in the end point and send mocked command data to the client-side.

Plan for Evaluation / Demonstration

Originally, we planned to do a test demonstration on the project by recording a group member using the object detection and animal detection software on a pet dog or cat, because CNN has been trained to identify both animals. However, after changing the scope of the project, it can be demonstrated online, since we can deploy the web API. For now, since the program is still its functionality divided, we can only test if each function works successfully. In order to test if each component of the project is functioning successfully, the CNN will be tested by using different test data sets and checking if the accuracy score is consistent. The object detection will be tested by using different video feeds and see if the program keeps track of the object of interest consistently. After merging the programs together we can then test the program, perhaps with a live video feed or other pre-recorded footage.

Objectives / Milestones

- Create and train on a new dataset created specifically for our purposes. We have created a dataset consisting of 7 classes, and each with about 1000 images.
- Achieve an accuracy rating of 70% or higher on our CNN (achieved). By using the pre-trained CNN model, MobileNet V2, we were able to have ~95% accuracy in our dataset.
- Combine the pieces of our project into a single entity, and test it to ensure that it works correct.
- Creating a web server to store all of our data and have our application work remotely
- Deploy the software suite and demonstrate the client-side (or start adding on hardware applications for demo purposes if we finish the software)