Project: Counting Your Chickens

▼ 1. Background knowledge:

1. chickens and coop in yard
2. 4 cameras on the corners of the yard fence posts looking inward toward the coop.
3. Cameras are from varied brands and resolutions is 480p+

Current uncertain factors:

1. What is the size of the yard and coop, and the amount of chicken
2. Is it a see-through coop or not?
3. The position and angle of the cameras? where is the view intersection in the scene?
4. What is the FOV of the camera, lens of the camera
5. Types of the camera(depth camera(lidar or image-based), color camera, infrared camera, etc)
6. What is camera recording frame rate.

▼ 2. Main Goal

The main question(goal) could be broken down as follow:

1. How to identify and track chicken by using ML/CV solution

2. How to track chickens around the yard.
3. How to know if chickens are inside/outside the coop.
4. What sort of architecture it would require,
5. what are the potential downfalls of the solution(1~4).

Before diving into a solution, we should review the **Background knowledge**, if some conditions are uncertain, we may need to provide assumptions for the potential solutions.

For question 2.1

The deep learning-based (instance)object detection (mask R-CNN, YOLO, etc) would be a good method to detect multiple different chickens from the image or image sequence, and it is capable to run in real-time, which is very good to identify the fast moving objects like chickens. If there are too many chickens and they occluded with each other, we should rise up the camera and tile the angle down a bit, and it will get better detection performance.

Potential downfalls:

The detection algorithm may need a lot of labeled data and may be costly or time-consuming to get good data for training and testing.

Rising up the camera position, may reduce the clarity of the object, e.g there are fewer pixels(or texture)for each chicken on the image, be aware of the change especially when the camera's native resolution is low, or the video stream running on higher speed(the resolution of the frame maybe become smaller). Which will bring in another factor that will decrease the detection performance.

For question 2.2

Once we have chickens detected, tracking each independent chicken around the yard is straightforward. We can use Kalman filtering or a deep learning-based approach SORT (Simple Online Real-time Tracking) with some reference points from the detection, to track multiple objects concurrently, These methods are also able to handle out-of-the-image and occlusion cases.

Other than the tracking algorithm, we also need to consider the FOV, position, and angles of these 4 cameras, because one camera may not cover the whole area of the yard. We need to stitch the synchronized images from these 4 cameras. Therefore, tracking can play smoothly among these cameras.

Potential downfalls:

It needs a consistent video stream system and a reliable detection algorithm.

External movement may change the camera's position and affect the stitching image.

Having Synchronized images is not easy, if one of the cameras fails, the tracking may be ruined.

For question 2.3

We may have two types of the coop.

If it is a see-through coop(Fig.1).

3.a If we have a depth map(directly from a depth camera or calculate from the color images same time from different angles) that covers the coop area, then we can align the depth map and tracking information plus the boundary of the coop to determine if the chicken is inside or outside the coop.

3.b If we are not able to achieve the depth map, we can use a deep learning-based activity recognition algorithm, consider the chicken and the coop entrance as the main factors, and train getting-in and getting-out as two different classes. The activity recognition algorithm will send us a signal about which one of these events(getting-in or getting-out) occurred. Therefore we know whether a chicken is inside the coop or outside the coop.

Potential downfalls:

To be able to let the activity recognition algorithm perform well, we need an obvious door(and door frame), and only one chicken in/out at the entrance.

And adding lasers or narrow-angle infrared sensors, on the door to detect the getting-in/getting-out, could be cheaper and more effective that the activity recognition algorithm.

Method 3.b works for the non-see-through coop(Fig2.) as well.

For question 2.4

In terms of architecture, it would mainly involve cameras, (battery)power cords, data transmission devices(e.g. WIFI), computing systems, data storage, and a software algorithm(computer vision + deep learning) implementation environment(CPU+ (GPU or TPU)).

Since electronic devices are vulnerable in the outdoor environment, I would minimize the pieces of equipment needed in the yard. Ideally, large capacity battery + camera + WIFI + waterproof housing, Focus on reliably collecting and transmitting data.

Potential downfalls:

We may need to change batteries periodically. However, it can avoid many disadvantages of using the power cord.

I would recommend to setup an on-premises computing system and data storage to run the algorithms and handle the data, instead of the cloud-based solution. Because streaming video will take a lot of bandwidth and need a very reliable network, otherwise the algorithm will not work well. On the cost side, the cloud-based solution will be more expensive in a long run.

Potential downfalls:

It will have the risk of the local machine failure, if we do not have a backup, all data will be gone, however, the cloud-based solution has better framework to counter the regional natural disaster and local machine failure.

▼ 3. Additional Questions:

▼ Questions

1. What if we want to keep an eye out for predators like foxes that might try to get into the coop?
2. What if we moved the coop around the yard? Would that change anything about the ML/CV solution?
3. What about the position of the sun or various weather conditions that may arise?
4. What if we wanted to port the solution over to other wildlife like ducks in a pond etc.? What would we need to change/think about?

▼ Answers

▼ The answer to question 3.1

One of the advantages of using deep learning-based (instance)object detection is that we can train the algorithm to learn new objects(predators) easily if we have enough labeled image data about the predators. And images with foxes are not hard to find.

Once we have the model trained, it can detect chickens and foxes simultaneously, and then track them.

▼ The answer to question 3.2

It does not affect the detection and tracking algorithm too much, but it needs some image(camera related) preprocessing to accommodate the changes.

For example, the coop in a new position may block the views of the camera, or the entrance of the coop is around the blind zone of the cameras, or the coop is too far from the camera and the camera's native resolution is not good enough to identify the object in the near area of the field.

Based on the new location of the coop, we may need to adjust the camera angle, swap the camera location, etc. In general, the input image for detection requires a smaller image resolution than the input camera provided, in this case, we can crop the image from the camera instead of scaling down and then feeding it into the detection algorithm. This type of preprocessing can help to gain more pixels(texture) for the long distant object.

▼ The answer to question 3.3

Those are the common challenges for camera imaging.
Direct sunlight may cause overexposure on the camera sensor. It depends on the sensor type(CCD or CMOS), and the angle and intensity of the sun, the whole or partial image may be pure white, and the image may be useless then. Nowadays, the newer camera has better processes on overexposure(pixels are with high value but not the max)issues. We can try image equalization to see if we can reduce this effect of overexposure.

We can consider rain as one type of noise, and it is moving fast, by using some average method algorithm or multi-camera strategies, we can reconstruct better images. Raindrops may stick on the lens which will block the camera view.

Fog is hard to deal with in general, it depends on the density of the fog, deep learning-based algorithm can handle a bit of light fog if fog augmentation images are added during training.

When snow turns into ice, it may be very reflective.

▼ The answer to question 3.4

First of all, targeting wildlife means this is no longer a closed-loop environment, ducks may not stay in the scene, the pond could be very big, and different type of wild ducks may look very different or other wildlife looks like a duck, e.g. a goose.

We need to add more cameras if the pond is bigger.

It may be hard to get enough good images of wild ducks for training, and we need more environment data for the model to reduce wild environment noise.

We need to add counter classes to duck, to suppress the false detection.

Enhance the tracking algorithm that could be able to track objects across multiple cameras.

▼ 4. Appendix

1. coop types

Fig.1 see-through coop

Fig.2 Non-see-through coop