Machine Learning Engineer Nanodegree

Capstone Proposal

Rahim Kassanaly
May 23, 2019

Proposal

Domain Background

Drones are impacting our society in may ways. We can now accomplish tasks that weren't possible before or that required a lot human intervention. They have completely changed the way we take pictures or we gather information of remote areas. Most of the drones are human controlled but with the rise of artificial intelligence we are entering a whole new world. I am passionate about robotics and autonomous vehicle. I possess a programmable drone and I have always wanted to use it to create an AI project. I find it astonishing to observe an autonomous vehicle making its own decisions based on his environment.

Problem Statement

Using my Tello drone my goal is to create a software that will allow it to track and follow an individual. The drone is equipped with a 720p camera that streams over Wi-Fi.

They are two problems that need to solved:

- Object detection: Ability to detect presence of an object AND location within the picture.
- Drone Tracking: Once the target is acquired, we need to control the drone accordingly to track the detected object. In others words, based on the location of the object in the picture we need to move the drone to the proper direction.

Datasets and Inputs

Since my goal is to detect a human in a picture, I choose to use VOC and COCO datasets. These are popular datasets that contains human pictures and annotations that are required for object detection.

VOC and COCO dataset will be use to train the model.

The drone camera frames will then be fed to the neural network to obtain object detection boxes. Based on that the output will be commands to the drone to move accordingly.

Solution Statement

- Object detection: There are many implementations that allows this, such as YOLO, SSD, R-FCN, etc. All those implementations use convolutional networks combined with a classifier. I choose the YOLO algorithm since it is faster and required less resources.
- Drone Tracking: I would like to solve this problem using reinforcement learning. As I did in the nanodegree drone project, I will use the DDPG algorithm to solve this continuous space. I have a target position of the center of the object in the picture and the current centroid point of the drone in the picture. Based on these variables we can use DDPG to converge to this position using an optimal policy. In the case that there are multiple objects detected in a frame, the drone will arbitrary choose one and track it.

Benchmark Model

I will compare the model performance to the performance if I would pilot the drone myself to center it to the object of interest. This way I can estimate of bad the model is performing comparing to a human agent.

Evaluation Metrics

I will use two metrics to determine the solution model:

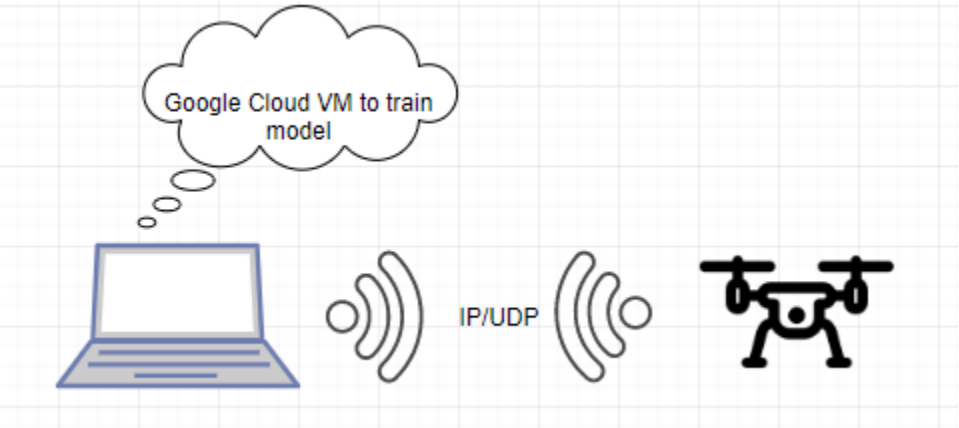
- Accuracy of the drone position in regards to the center of the object. A squared distance between the coordinate of the drone and the target.
- Time to converge to the target.

Project Design

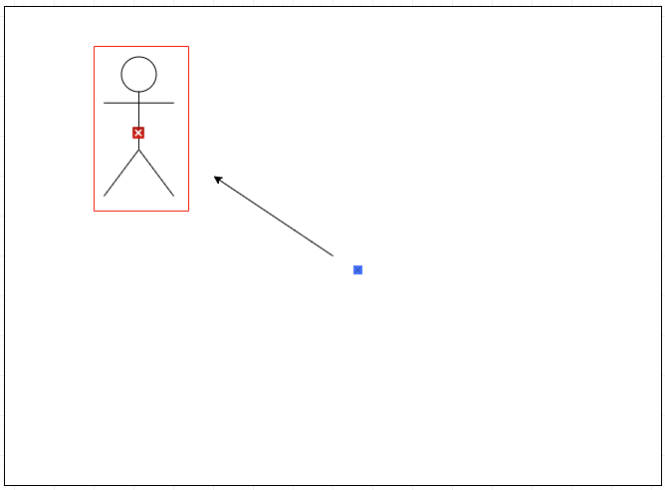
The drone is communicating via Wi-Fi using UDP protocol to send video stream and receive commands.

Since training the model is really demanding in term of resource, I will train the model on Google Cloud VM if needed.

Please note that the model will be running and processing frames from the laptop. Commands are sent from the laptop to the drone via UDP.



General workflow of the system:



1. Frame is sent from the drone to the laptop.
2. Frame is fed to the object detection model running on the laptop.
3. Bounding box of the object is returned with pixel coordinate of its center
4. The RL agent on the laptop will send commands to the drone so the blue point gets as close possible to the red one. Please note that commands to the drone are limited to the following: rotate clockwise, rotate counter clockwise, move

forward and move backward. There are three observations to define a state: x position, y position and size of the box which indicated the distance from the object.

5. When target is reached, restart from step 1.

References

COCO: http://cocodataset.org/#download

VOC: http://host.robots.ox.ac.uk/pascal/VOC/index.html