# Deniz Kartal (deniz001) - Public Page

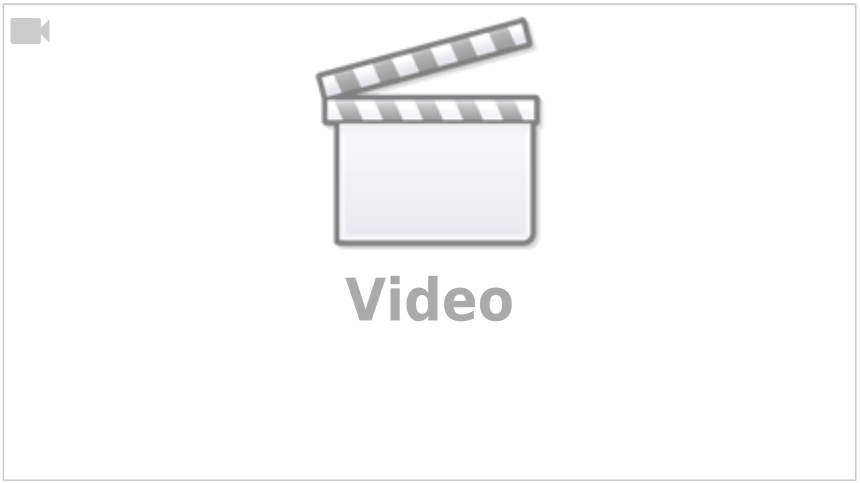
## Drone Tracker

### Background

### Components

| Component | Brand, Type | Datasheet | Link |
|---|---|---|---|
| Pan and Tilt Unit | FLIR PTU-5 | PTU-5-USER-MANUAL<br>FLIR PTU-5 DATASHEET | PTU-5 @ FLIR |
| Jetson Xavier NX | NVIDIA | Getting started with Jetson Xavier NX developer Kit | Embedded Computing with Jetson |
| Camera | | | |

### Pan and Tilt Unit




Source: FLIR Systems Inc. YouTube Channel.

Thread: 1/4-20 UNC THRU (from mechanical drawings in the user manual)

### 0. Introduction

On this page, I would like to introduce an object tracking system. Main idea is to follow a drone indoor but the system can be trained to track any object.

The system consists of a pan and tilt unit, a camera, and software.

Software is implemented to:

- detect the drone object or let the user select a bounding box around the drone in the first frame.
- find/track the object in the future frames.
- control the PTU(Pan and Tilt Unit) using a PID controller to put the drone object in center of the

frames.

There are two options in order to have a working system that tracks a drone. The first option is to basically select an initial bounding box and, the second one is to train a neural network that does the manually bounding box selection step automatically using the trained CNN. Both of those mentioned solutions are implemented in software.

The idea is very simple, we first “detect” or “select” a bounding box around the drone object, then use a tracking algorithm to track the drone object in each frame, and then control the PTU using the PID controller to put the drone object into the center of the frames.

***All the source code for this project and usage demo can be found here:***
https://gitlab.com/poseidon42/object-tracker

## 1. Data Collection

Data is very important if we use the second option that we train a neural network. There are a lot of resources to load datasets but I could not find an efficient one for drone images therefore I have written a python script that downloads a good number of user specified object images from google to create my own datasets. The script can be found in the project.

## 2. Image Augmentation

Data augmentation aims to increase the accuracy of the training by creating different versions of the images that we gathered using the python script to collect our data. This way we can use the original images in the validation step and use the augmented images in the training step. Again the python script can be found in the project folder.

## 3. Object Detection

In object detection we aim to find an instance of an object of a certain class in an image or a frame of a video stream and output the bounding box coordinates of that instance. In our case, one of our drone from our laboratory is an instance of the drone object.

Object detection can be achieved using machine learning and deep learning based algorithms. In machine learning approaches we need to feature engineer in order to define features. While in deep learning, we do not need to manually define the features but rather the CNN(Convolutional Neural Network) finds its way to define the features.

Example of Machine Learning approaches:

- Viola-Jones object detection framework based on Haar features
- Scale-invariant feature transform (SIFT)
- Histogram of oriented gradients (HOG) features

Example of Deep Learning approaches:

- R-CNN(Region-based Convolutional Neural Networks)

- Fast R-CNN
- Faster R-CNN
- SSD (Single Shot MultiBox Detector)
- YOLO (You Only Look Once)

In this project, I have decided to try out both YOLO and SSD, but in the future faster R-CNN could be implemented as well.

#### Faster R-CNN:

#### YOLO:

YOLO is a real-time object detection algorithm. YOLO divides each image into a grid of S x S and each grid predicts N bounding boxes and confidence. The confidence reflects the accuracy of the bounding box and whether the bounding box actually contains an object(regardless of class). YOLO also predicts the classification score for each box for every class in training. In total S*S*N boxes are predicted. However, most of these boxes have low confidence scores and if we set a threshold say 30% confidence, we can remove most of them. Image is run through the CNN only once at run time. Framework for YOLO is also YOLO.

#### SSD:

SSD runs a convolutional network on input image only once and calculates a feature map. SSD uses anchor boxes at various aspect ratio and learn the off-set rather than learning the box. In order to handle the scale, SSD predicts bounding boxes after multiple convolutional layers. Since each convolutional layer operates at a different scale, it is able to detect objects of various scales.

#### Why not using tracking by detecting in each frame?

- There can be multiple objects entering and exiting the scene of the camera over time in frames, in that case there is no possibility to match or connect the objects in the current frame with the previous frames that the camera was recording in the past.
- The object may suddenly go out of the camera's view in the next frame then another same type of object may get in the frame, in this scenario there is no way that the system can figure out which object that the system was actually tracking. To sum up, there is no way for the system to have an idea about object's current and past movements but in this project the purpose is to track a unique object so that we can calculate where the object goes and create a motion map.
- There can be blur or noise in the frames due to the motion of the object or camera, that is why the object may look very different so the detection would fail again.
- The object may have a very viewpoint that trained neural network was not prepared for that or the object may go far away so that the detection is not available(as the object gets far from the camera's view there will be a huge change in the scale of the object this would cause a failure in detection).
- Low resolution, the number of pixels
- Tracking algorithms are faster than object detection because trackers do not intend to learn all the detailed information of the object like the detection algorithms.(In the future when we have

much more powerful hardware we may go for detection in each frame rather than tracking!)

## 4. Object Tracking

By object tracking we can uniquely identify an object instance, so the drone instance.

The goal of an object tracker is to locate a moving object by estimating the location of the target object in the future frames and check if the object in current frame is the same as the one which was in the very previous frame.

Tracking process goes by first initially defining a bounding box of the target object.

#### Motion modelling

Objects do not randomly move in the space but rather they have moving characteristics and patterns which can be modeled. movement prediction model to remember how the object moved in the past frames so that we can predict the next possible location space of the object. An object tracker tries to understand and model the motion of an object mostly in the pixel level, that is called the motion model. it can estimate the location of an object in the future frames that would reduce the size of the image that the tracker looks for the object.

#### Appearance Modelling

A good tracker must understand the appearance of the object that the tracker tracks, they must learn to differentiate the object from the background which is in the image.

#### Motion Detection

A good tracker must learn to estimate the motion of the object in order to have a guess about the space that the target possibly can be present in the frame.

#### Object Localization

Focus the attention on the region of interest in the frame. Data reduction A good tracker uses the motion estimation and figures out the possible region where the target may be locating in the current frame and scan this area using the model that the tracker created about the object's appearance in the past frames and finally find the exact location of the target in that current frame.

Offline trackers are used when we have a recorded media, in that case we use also the future frames to make tracking predictions. While online trackers can only use the past frames to model the appearance, and the motion of the object for tracking estimations.

Online learning trackers train itself to learn about the object(which is initially selected and the bounding box is inputted to the tracker for learning) using the array of frames that start from the initial frame till the frame that is one before the current frame.

Offline learning trackers are trained offline and they do not learn anything during the tracking process. An offline tracker may be trained to identify an object before the tracking starts.

Most of the traditional trackers that are available in OpenCV are not based on Deep Learning. (KCF is the best one)

CNN(Convolutional Neural Network) based offline trackers: GOTURN CNN(Convolutional Neural Network) based online trackers: MDNet(Multi domain network) best DL based

Tracking algorithms available:

- **Boosting Tracker:** A real-time object tracking based on a novel online version of the AdaBoost algorithm. The classifier uses the surrounding background as negative examples in update step to avoid the drifting problem.
- **MIL Tracker:**
- **KCF Tracker:**
- **KCF Tracker:**
- **KCF Tracker:**
- **KCF Tracker:**
- **KCF Tracker:**
- **KCF Tracker:**

## 5. PID Controller

PID controller is a feedback control loop, its output is fed back to the system as an input to reach the target.

The output of the tracking algorithm is a bounding box that represents the location of the object that we track, that is the drone object. Using the output of the tracker the error, that is the distance between the center of the current frame and the center of the drone object in the current frame, is calculated and this error is the input to the PID controller which tells the PTU(Pan and Tilt Unit) in which direction to move in order to put the object in the center of the current frame.