Overview

Goal

An Object Detection and Tracking system that can run on the edge, at a reasonable framerate, that is low cost to develop.

Challenges include:

- Video (computationally expensive)
- Object detection (need to identify objects in frame)
- Object tracking (need to track objects from frame to frame)
- Small objects (hopefully can get 10px)
- Moving background (clouds)
- Multiple moving objects
- Moving foreground (rain on lens)
- Environmental conditions (weather, lighting, day/night)
- Unknown object features
- Runs on Edge

This makes the problem an absolute cutting edge Computer Vision one in 2021.

Approach

Use the historical data collected by the SkyHub software to train a DNN.
Have humans annotate (label) the videos to be used as input.

Historical Data

We have 64k MKV format video files stored in S3 totalling about 1.2TB of data.
The vast majority of this data is clouds or rain.
About 10% of it contains things like planes, birds, insects.

Annotation Proposal

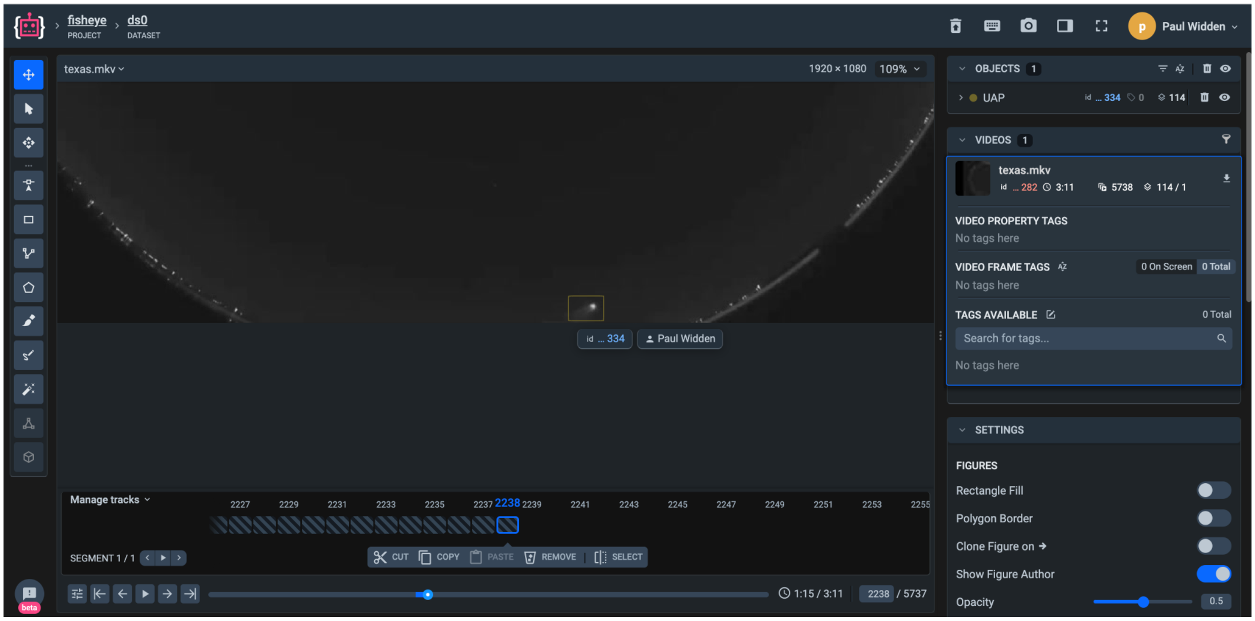
Build a pipeline consisting of two stages:

Stage 1: Take the historical data and make a first best guess at object tracks using traditional CV algorithms, splitting the videos into individual tracks and assigning a Tracking ID that is consistent throughout the video for a given object.
Stage 2: Upload the split videos and associated annotations to https://supervise.ly and have humans verify these tracks in the Web UI.

The stage 1 proposal is to use the existing traditional ODT POC as the basis for generating automated tracks in a JSON metadata file in a format supported by supervise.ly:

```
{
    "size": {
        "height": 1080,
        "width": 1920
    },
    "description": "",
    "key": "592a3fb03bef4040b6ac081ee95e6af3",
    "tags": [],
    "objects": [
        {
            "key": "99bdd15179c1454eb1bc04ece23e19dd",
            "classTitle": "UAP",
            "tags": [],
            "labelerLogin": "pwidden",
            "updatedAt": "2021-10-12T03:36:45.256Z",
            "createdAt": "2021-10-12T03:36:45.256Z"
        }
    ],
    "frames": [
        {
            "index": 2125,
            "figures": [
                {
                    "key": "3d2283ac7f1d419aa3c855de49a345a8",
                    "objectKey": "99bdd15179c1454eb1bc04ece23e19dd",
                    "geometryType": "rectangle",
                    "geometry": {
                        "points": {
                            "exterior": [
                                [
                                    1039,
                                    1076
                                ],
                                [
                                    1044,
                                    1079
                        ]
                    ],
                    "interior": []
                }
            },
            "labelerLogin": "pwidden",
            "updatedAt": "2021-10-12T03:37:29.651Z",
            "createdAt": "2021-10-12T03:37:26.390Z"
        }
    ]
},
…
```

And then to upload this metadata along with the videos to supervise.ly where we can get volunteers to help with the verification of the tracks:



The approach would be to do this in small batches until we had enough data to train the model.

Current State

There's a POC for using a traditional object detector (SimpleBlobDetector) coupled with a fairly robust tracker (CSRT), which outputs videos like this: https://drive.google.com/file/d/1A5SoMTSbSrd-qZjHnTFBzWTnHyH7Vs1S/view?usp=sharing

N.B. The main reasons we can't use this in the edge ODT is that it's not very good at determining the difference between clouds and flying objects and it's not very performant.

The tracker can be found here (contact UAPPaul for access):
https://github.com/sky360/simpletracker

The POC takes as input a directory containing videos in mkv format. It produces as output three directories:

- Movement: videos from the input directory that were identified as having movement are moved here.
- No_movement: videos from the input directory that were identified as having no movement are moved here.
- Transcoded: Videos with movement are transcoded to mp4 and bounding boxes rendered into them.

Stage 1 Tasks Remaining

https://github.com/Sky360-Repository/simpletracker/projects/1

ML Model (Fisheye)

My current approach is to try and see if we can train any network to recognise moving objects in the sky whilst ignoring (moving) background (or in some cases foreground as the object goes behind cloud).

I'm taking videos from my fisheye and running them though the simpletracker using the STF output. Then I'm excluding any tracks that are background and marking every other flying thing as 'unknown' - planes, birds, insects etc. I'm doing this because I don't really care what the object is (no need to classify at this stage - that comes later on the PTZ).
The output of the STF contains image frames with 3 color channels and one alpha channel representing the output of an opencv backgroundSubtractorKNN for the current frame vs previous ones, thus incorporating the temporal aspect of motion. I'm hoping this will give the NN a good indication of where to focus when looking for an object. Another approach would be to use an RNN.

The model I've chosen to start with is the retinanet, primarily because it has a fairly good balance of performance and accuracy and it was relatively easy to get up and going on. I've modified it to take a fourth alpha channel and am about to start training.

Pipeline

RGBA Image frames output from simple tracker and converted to CSV format using stf2csv.py
Normalize
Augment with some flipped images

Concerns

- CNN performance on small objects (consider smaller anchors?)
- Nano processing performance at inference Benchmarks for Nano (note image sizes)
- Large input images (MobileNet looks at images 200*200, we're looking at 2880x2880)

People

We're looking for people who can help develop this system, specifically Python, ML and DevOps.

Paul: Python programmer, background in Computer Vision from years ago, catching up on latest developments in Deep Learning.

Mike:

Appendix

Resources

New to Neural Nets/Deep Learning, start here:
https://www.manning.com/books/deep-learning-with-python-second-edition

Learn Deep Learning in Computer Vision:
https://www.manning.com/books/deep-learning-for-vision-systems

SOTA Object Detection and Tracking:
https://blog.netcetera.com/object-detection-and-tracking-in-2020-f10fb6ff9af3

History

Decisions

Terminology

ODT: Object Detector and Tracker
There are lots of different versions of ODT's, some use traditional Computer Vision model and some use more modern Machine Learning models.

CV: Computer Vision
The field of computing that deals with processing and interpreting visual data. This encompases traditional methods and modern Machine Learning.

ML: Machine Learning
The study of computer algorithms that can improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence

DNN: Deep Neural Net
A large Neural Net with many layers and typically millions of weights.

SOTA: State Of The Art
Current, best in class solution

Fun

MobileNet

- Has 1.5M parameters (somewhat equivalent to synapse in the brain)
- Can classify about 1000 items

The Brain

- Has 1000 trillion synapses
- Costs $250k on average to raise a child in the US

GPT-3

- Has 175B parameters
- Costs $12m for a single training run