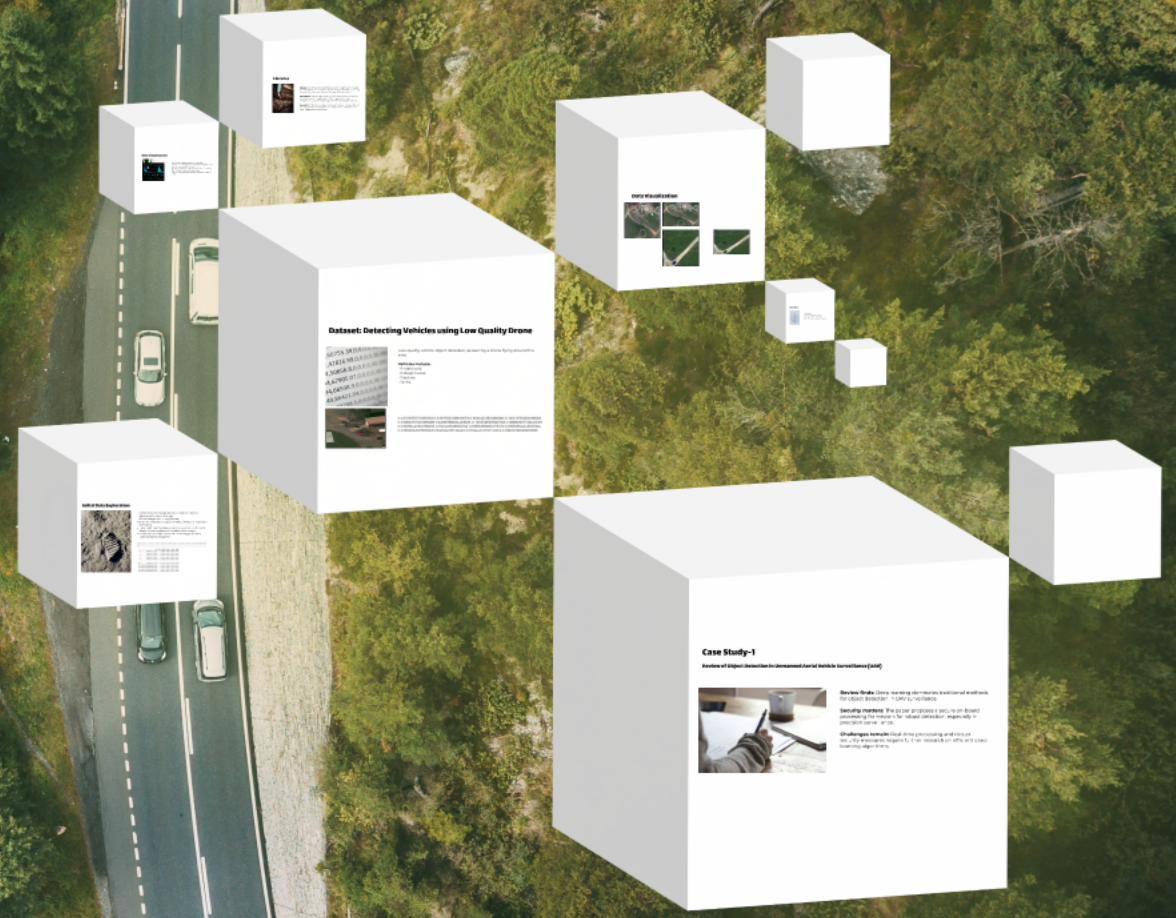
AeroWatch


Dataset: Detecting Vehicles using Low Quality Drone
Case Study-1

Case Study-1

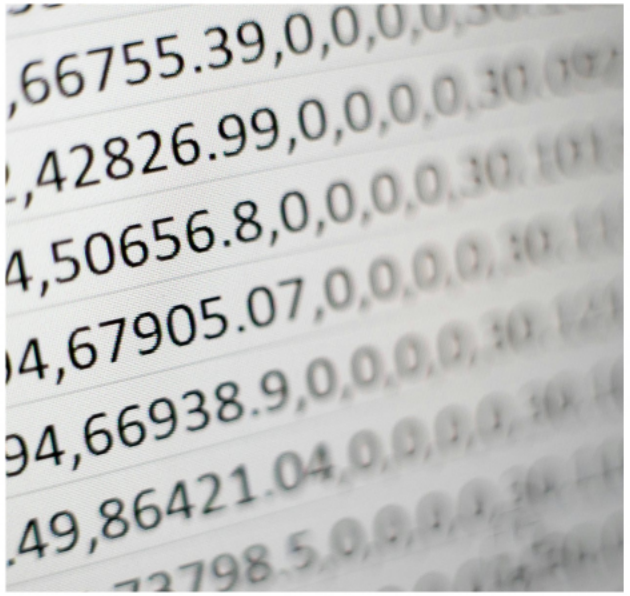
Review of Object Detection in Unmanned Aerial Vehicle Surveillance (UAV)

Review finds: Deep learning dominates traditional methods for object detection in UAV surveillance.

Security matters: The paper proposes a secure on-board processing framework for robust detection, especially in precision surveillance.

Challenges remain: Real-time processing and robust security measures require further research on efficient deep learning algorithms.

Dataset: Detecting Vehicles using Low Quality Drone



Low-quality vehicle object detec&on, as seen by a drone flying around the area.

Vehicles include:

- Private cars
- Pickup trucks
- Tractors
- Tanks

0 0.5123373710183938 0.4932705248990578 0.1094661283086586 0.10632570659488558
0 0.5850157021085688 0.6298788694481829 0.11305518169582768 0.08882907133243599
0 0.9035441902198295 0.762449528936743 0.09869896814715119 0.09555854643337824
0 0.9632122027815163 0.946164199192463 0.05114401076716021 4 0.08613728129205925

Initial Data Exploration

- Combining the Images and its Labels to create a dataframe fo easier Analysis
- All the images are in .png format.
- With the resolution output of 720p x 480p (i.e., Standard Definition).
- Labels defines the class, x-centre, y-centre, width, and height of the numbers of vehicle in the images.
- Created a new Dataframe with the images and the labels combined together.

```
df[['x_center', 'y_center', 'width', 'height']] = df[['x_center', 'y_center', 'width', 'height']].astype(float)

df
```

	file	class	x_center	y_center	width	height
0	f46fd5e4-out_2_3_230.png	0	0.066397	0.038358	0.084343	0.074024
1	f46fd5e4-out_2_3_230.png	0	0.100942	0.072678	0.112158	0.083445
2	f46fd5e4-out_2_3_230.png	0	0.063257	0.291386	0.085240	0.084791
3	f46fd5e4-out_2_3_230.png	0	0.071332	0.638627	0.079856	0.125168
4	f46fd5e4-out_2_3_230.png	0	0.396590	0.403769	0.089726	0.075370
...	...	...	...	...	...	...
1171	b23d330b-out_2_2_21.png	0	0.322118	0.871467	0.107672	0.122476
1172	56ef93a7-09152008flight2tape3_940.png	0	0.591297	0.928668	0.118439	0.061911
1173	56ef93a7-09152008flight2tape3_940.png	0	0.580081	0.964334	0.112158	0.055182
1174	56ef93a7-09152008flight2tape3_940.png	0	0.845671	0.885599	0.070884	0.080754
1175	56ef93a7-09152008flight2tape3_940.png	0	0.899507	0.888291	0.081651	0.104980

Libraries

- **Image:** This is the core library that deals with loading, saving, and processing images. It allows us to open image files (like jpg or png), convert them between different formats, and resize them.
- **ImageDraw:** Think of this library as your drawing tool for images. Once we have an image loaded with Image, we can use ImageDraw to add things on top of it, like lines, shapes, and text.
- **ImageFont:** This library deals with fonts. It allows to load different font styles and sizes and then use ImageDraw to write text onto your images using those fonts.

Data Visualization

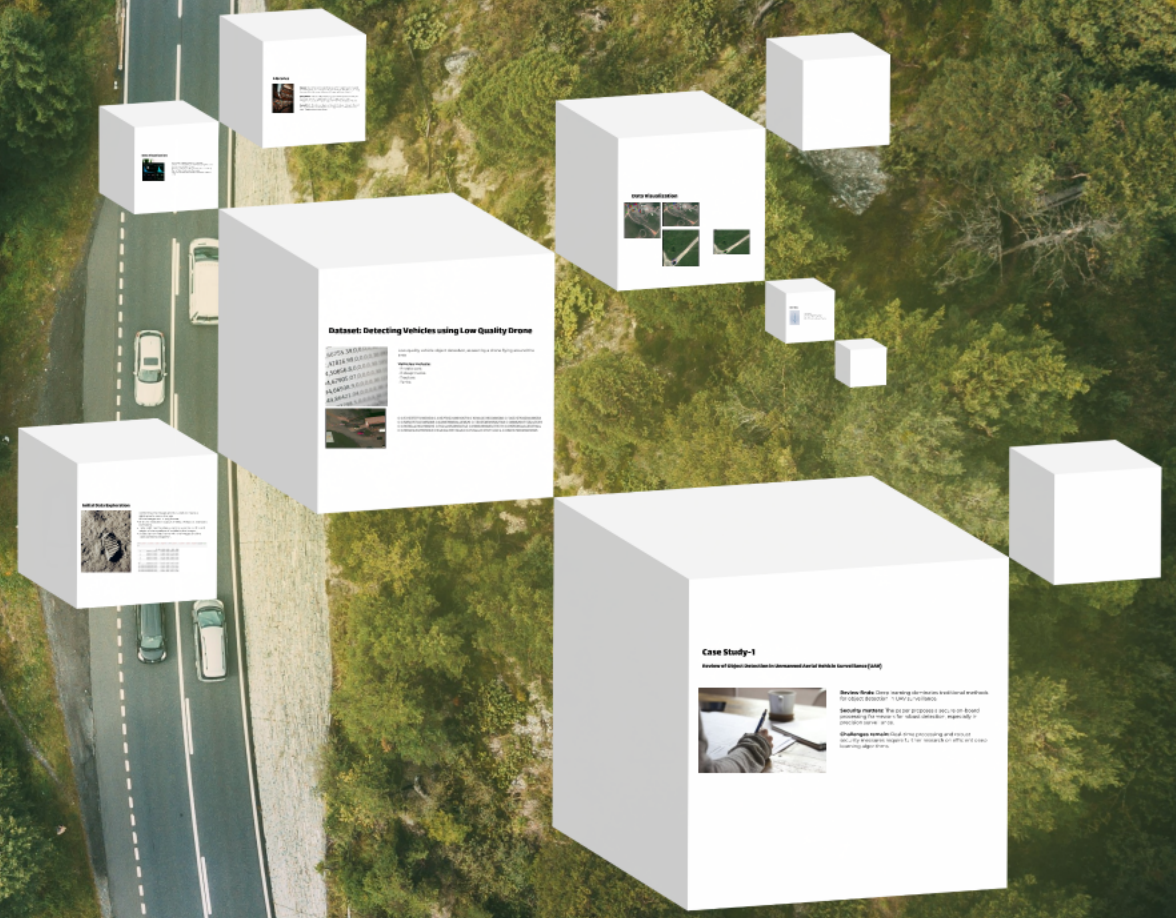
- Picking a random image for the dataset.
- Calculate the bounding co-ordinates using the labels for the coresponding images.
- Drawing the bounding box/outlines and calculating the number of vehicles in the image.
- For better uniformity, resizing the images to 320p X 320p.

Data Visualization

Next Step

- Training Setup.
- Models Selection: Simple Image. Classification, CNN, R-CNN, Yolo
- Model Training, Evaluation and Testing.

AeroWatch


Dataset: Detecting Vehicles using Low Quality Drone
Case Study-1