Vision-Based UAV Self-Positioning in Low-Altitude Urban Environments

December 31, 2025

1 Library setting

[ ]:
```
%pip install -r ../constraints.txt
```

[20]:
```
from datasets import load_dataset, Image, load_dataset_builder
from itertools import islice
from matplotlib import pyplot as plt
```

2 DenseUAV

Load dataset

[3]:
```
DenseUAV = load_dataset("Dmmm997/DenseUAV", split="train", streaming=True)
```

2.1 Dataset size

Storage size of each sample in the DenseUAV dataset

[6]:
```
import sys

# Get a sample
sample = next(iter(DenseUAV))
img = sample["image"]

# Get image dimensions
width, height = img.size
channels = len(img.getbands())  # RGB = 3, RGBA = 4, etc.

# Calculate size in bytes (assuming 8-bit per channel)
image_size_bytes = width * height * channels

# Convert to MB
image_size_mb = image_size_bytes / (1024 * 1024)

print(f"Image dimensions: {width}x{height}")
print(f"Channels: {channels}")
print(f"Image size: {image_size_mb:.4f} MB")
```

```
Image dimensions: 512x512
Channels: 3
Image size: 0.7500 MB
```

Simple calculation of the total dataset size

[14]:
```
sample_overall = 21845
```

Subset of the DenseUAV dataset

[22]:
```
dd = DenseUAV.shuffle(seed=42, buffer_size=1000).take(30)
```

2.2 Dataset testing

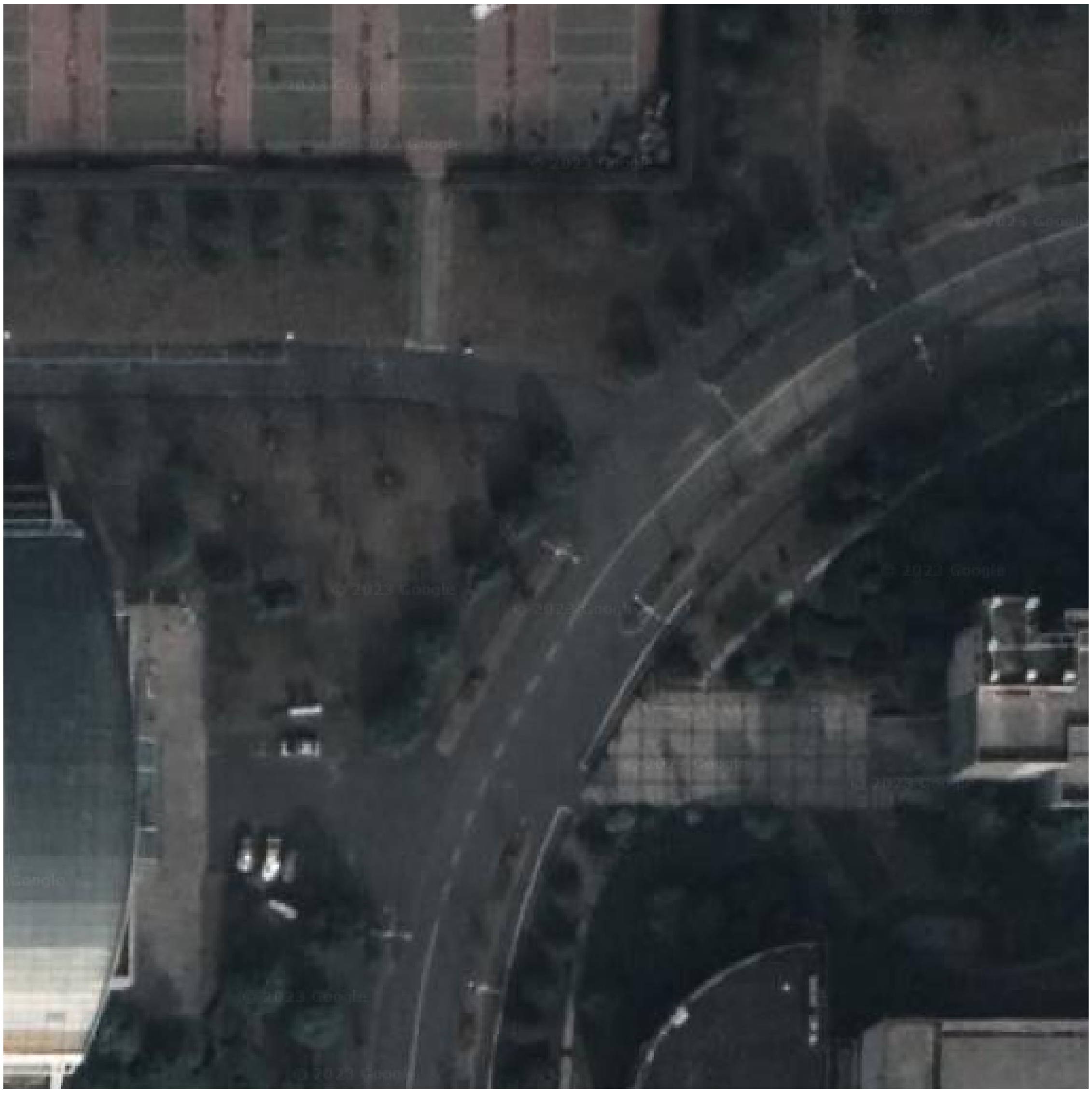
Check dataset features

[ ]:
```
dd.features
```

Get a first image from DenseUAV dataset

[26]:
```
imageSample = next(iter(DenseUAV))
img = imageSample["image"]
img
```

[26]:



List of 30 samples after shuffle with buffer size 1000:

[ ]:

```
for sample in dd:
    img = sample["image"]
    plt.imshow(img)
    plt.axis("off")
    plt.show()
```