Project: Parking Space Monitoring System

Sofia Gonçalves
Adrianna Domaszewicz

Project Overview

Goal:

Develop a system to monitor parking space occupancy using computer vision

Dataset:

Images of empty and occupied parking slots for training

Mask image to identify and locate the parking slots available

Parking lot video footage

EDA and Data Preprocessing

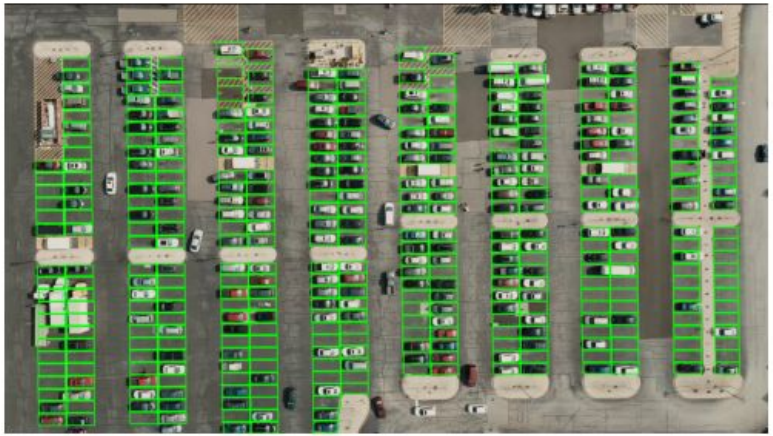
- Distribution of training data (e - 6091, o- 3045)
- Extracted frames from video
- Applied mask
- Data generators
- Normalize pixels values /255.0
- Convert images to numpy array
- Data types from ‘float64’ to ‘float32’
- Resized frames (64,64)(1280,720)
- Process data in batches

```
# Normalize pixel values
X = X_data.astype('float32') / 255.0
```

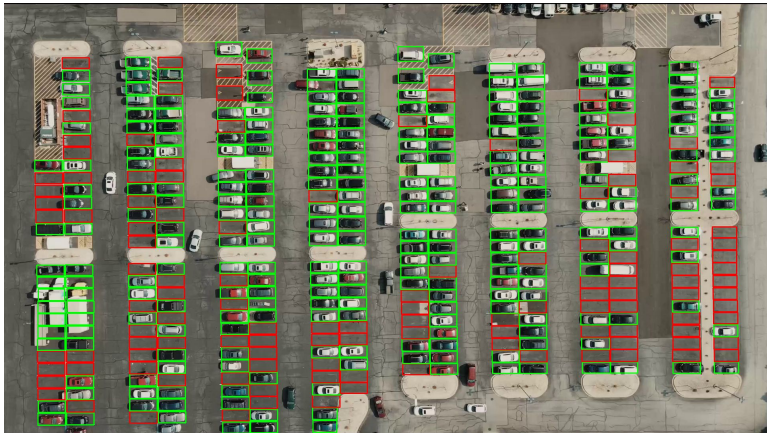
```
# Function to classify parking spots in a frame
def classify_parking_spots(frame, parking_spots, model):
    spot_images = [frame[y:y+h, x:x+w]
                   for (x, y, w, h) in parking_spots]
    spot_images_resized = [cv2.resize(spot, (64, 64))
                           for spot in spot_images]
    spot_images_resized = np.array(spot_images_resized).astype('float32') / 255.0
    spot_predictions = model.predict(spot_images_resized)
    return spot_predictions
```

First Frame of the Video

First Frame with Parking Spots



First Frame with Blacked Out Non-Parking Areas



Model Development

```
optimizer='adam',
loss='categorical_crossentropy'
metrics=['accuracy']
epochs=10
batch_size=32
validation_data=(X_test, y_test)
```

Model selection:

- YOLO (You Only Look Once)
- Logistic Regression
- Support Vector Machine (SVM)

Used: Convolutional Neural Network model

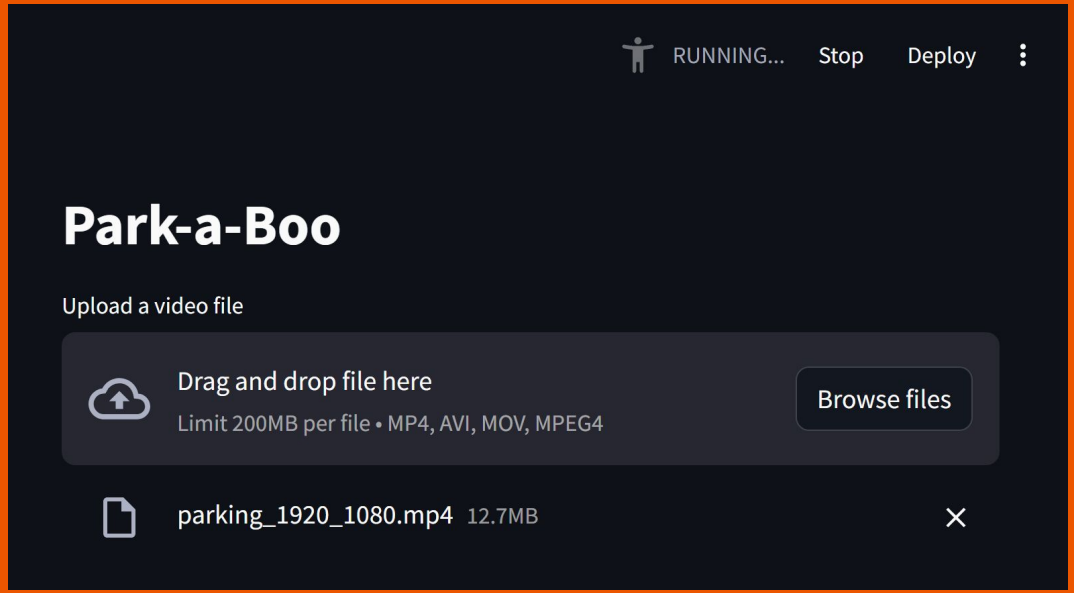
Trained on parking lot images (empty vs. occupied)

Tested on a video file

Labels 0 for empty spaces, 1 for occupied

Video output

App via Streamlit



Further Improvements and Conclusions

- Examine misclassified instances, where the model struggles (eg., gray cars)
- Features that are challenging for the model (e.g., poor lighting conditions, occlusions)
- Look for patterns and peak occupancy
- Test with different weather conditions other environmental factors
- Extract features from images using Histogram of Oriented Gradients (HOG)

Thank you!