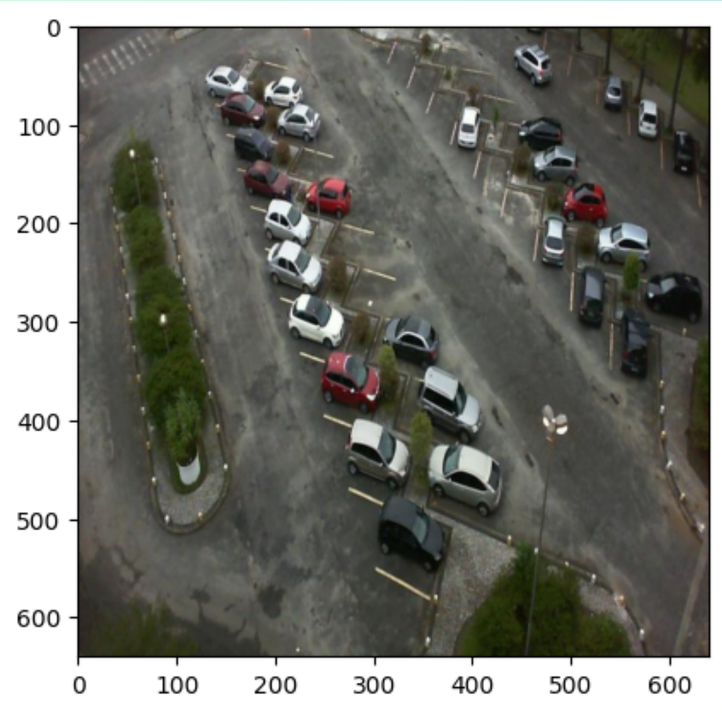
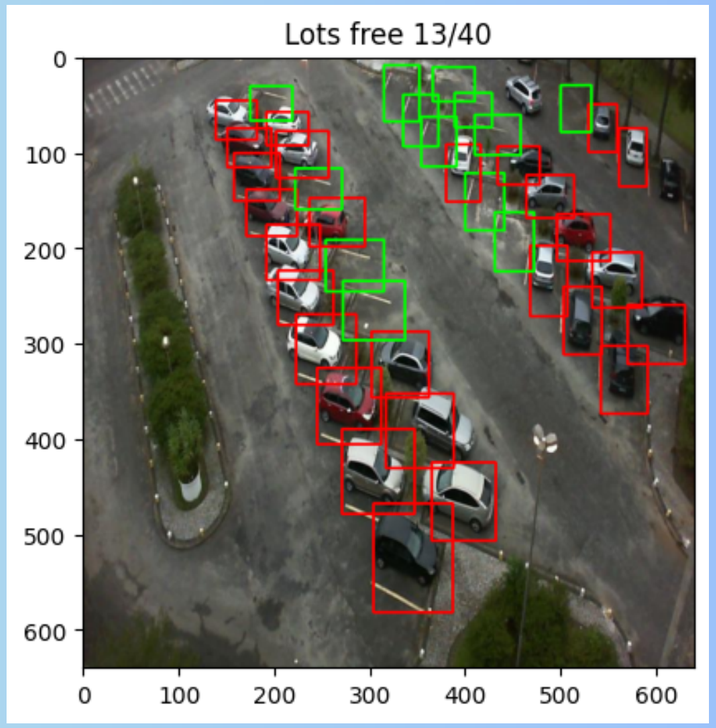
PARKING LOT DETECTION APP

Made by Ernest Knurov

PROJECT DESCRIPTION

Application for detecting free parking lots. Given parking image as input, app outputs image with highlited parking lots and statisctics of free lots





MAIN ISSUES

- How to find parking lots?
- How to find cars?
- How to determine that the car is in the parking lot?

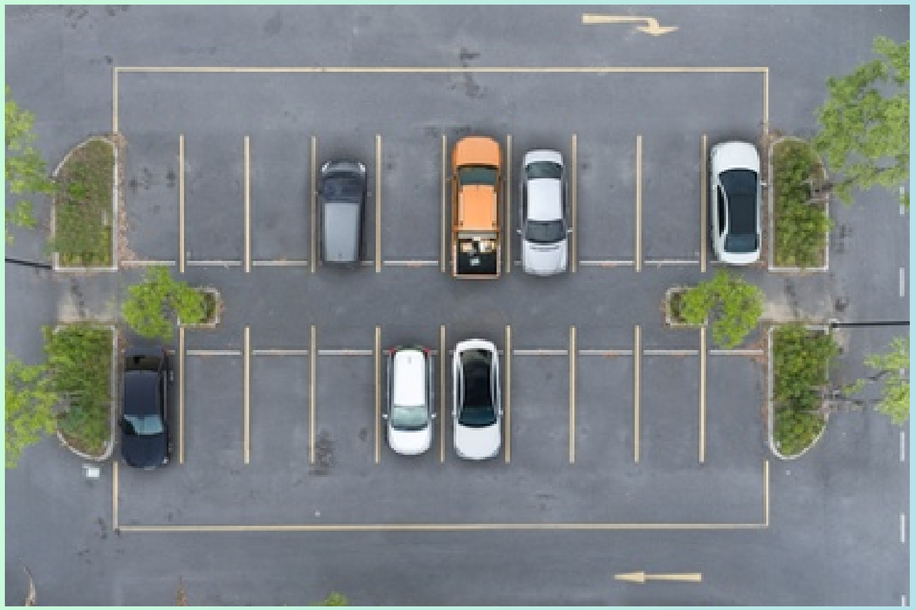
APPROCHOES TO FIND PARKING LOTS

APPROCHOES TO FIND PARKING LOTS

- Find lines between parking lots

APPROCHOES TO FIND PARKING LOTS

Find lines between parking lots



Unrealistic camera position

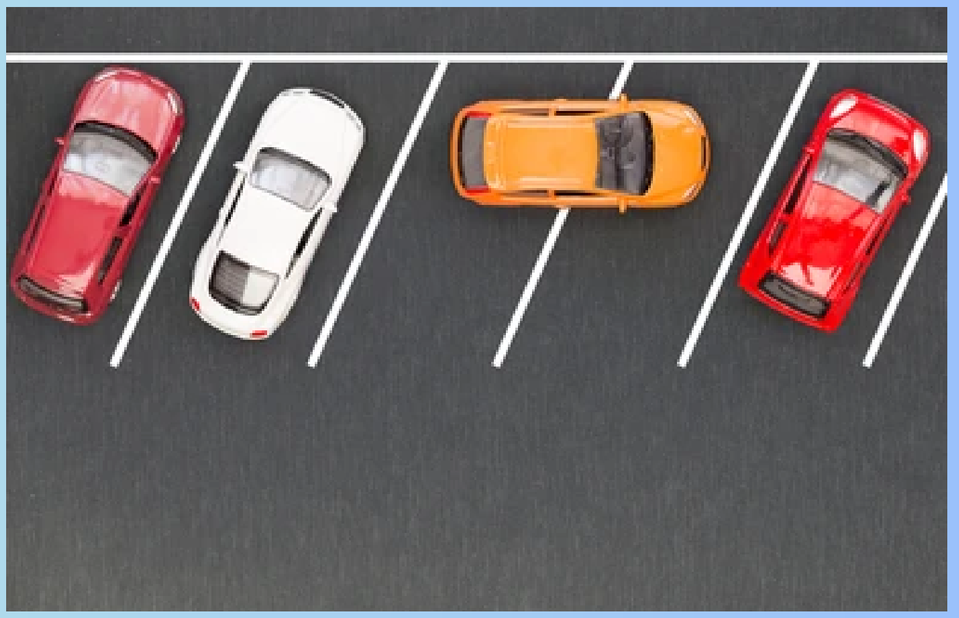
The lines are not visible

APPROCHOES TO FIND PARKING LOTS

> Assume that parking lots are places where cars stay for a while

APPROCHOES TO FIND PARKING LOTS

Assume that parking lots are places where cars stay for a while

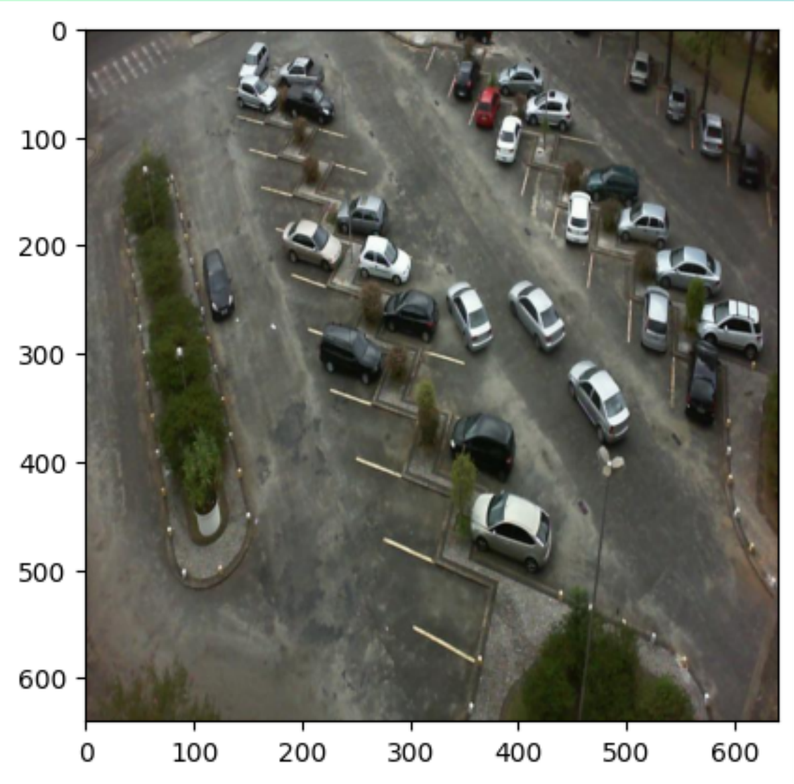


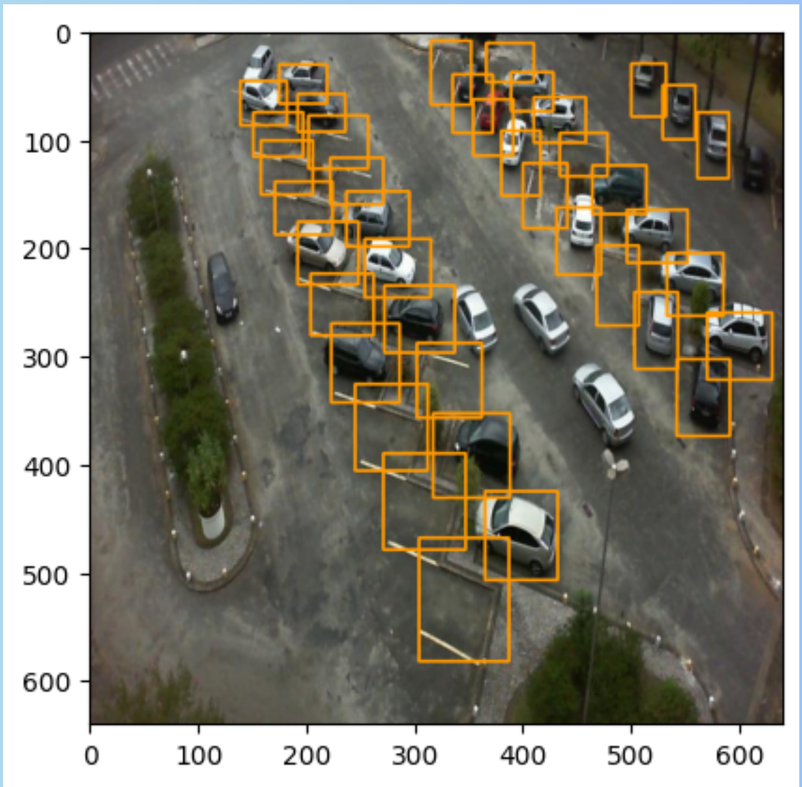
APPROCHOES TO FIND PARKING LOTS

- Mannually make the markup

APPROCHOES TO FIND PARKING LOTS

Manually make the markup





APPROCHOES TO FIND CARS

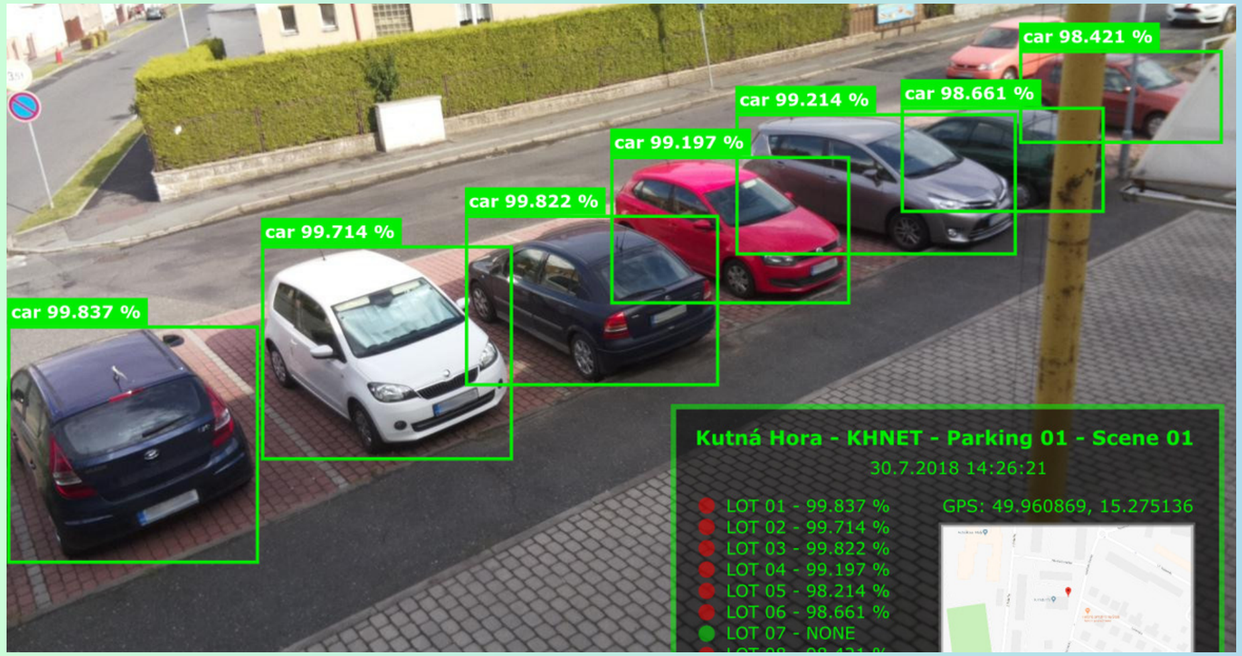
APPROCHOES TO FIND CARS

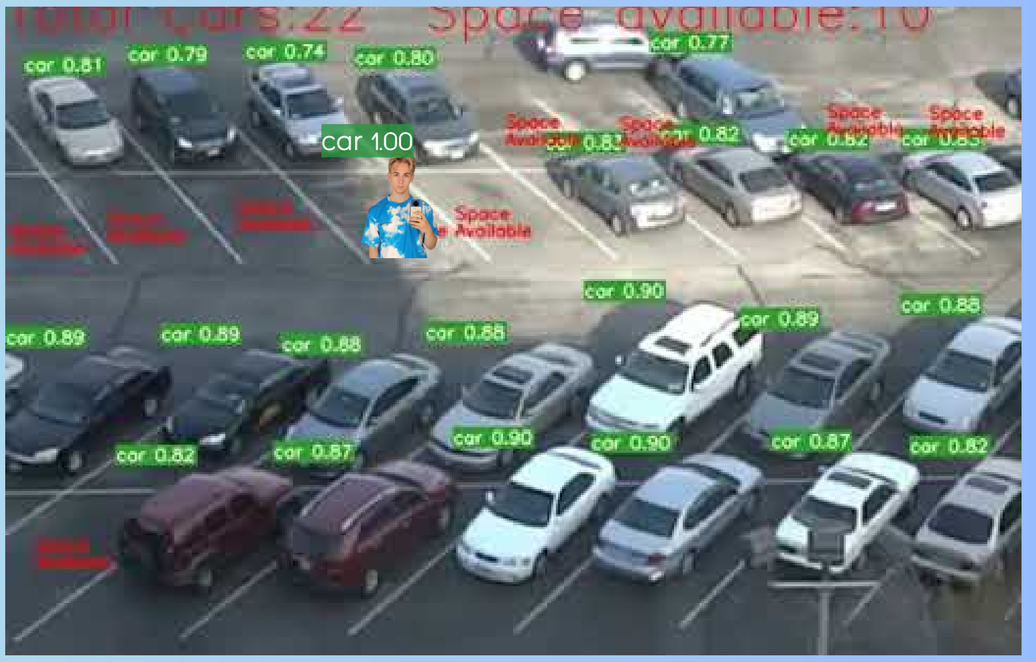
Use an object-detection algortihm on the entire image

APPROCHOES TO FIND CARS

Use an object-detection algortihm on the entire image

Examples:





APPROCHOES TO FIND CARS

> Use an object-detection algortihm on the entire image

Pros

- Can detect multiple objects on image
- Works fast, regardless of the number of objects in the image

Cons

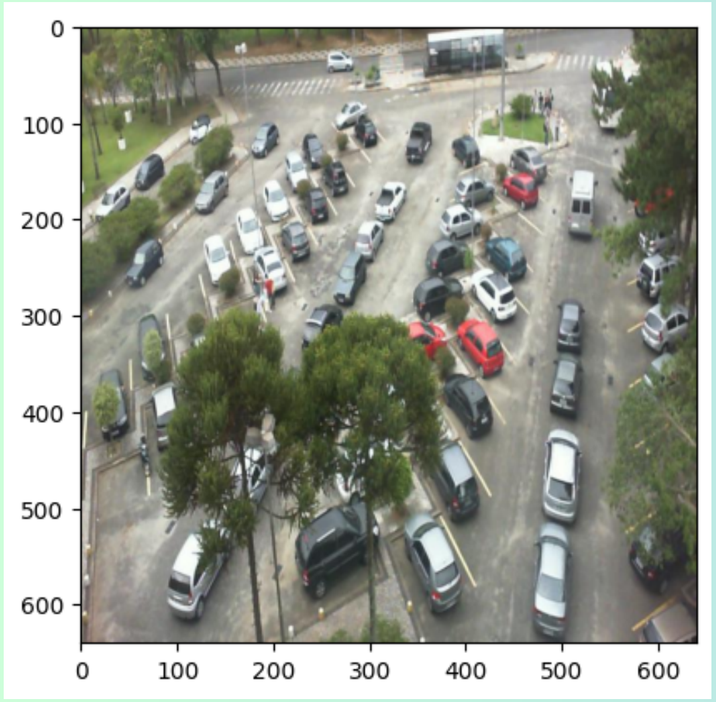
- Bad at detecting of small objects

APPROCHOES TO FIND CARS

Use classificator car/no car on cropped images of parking area

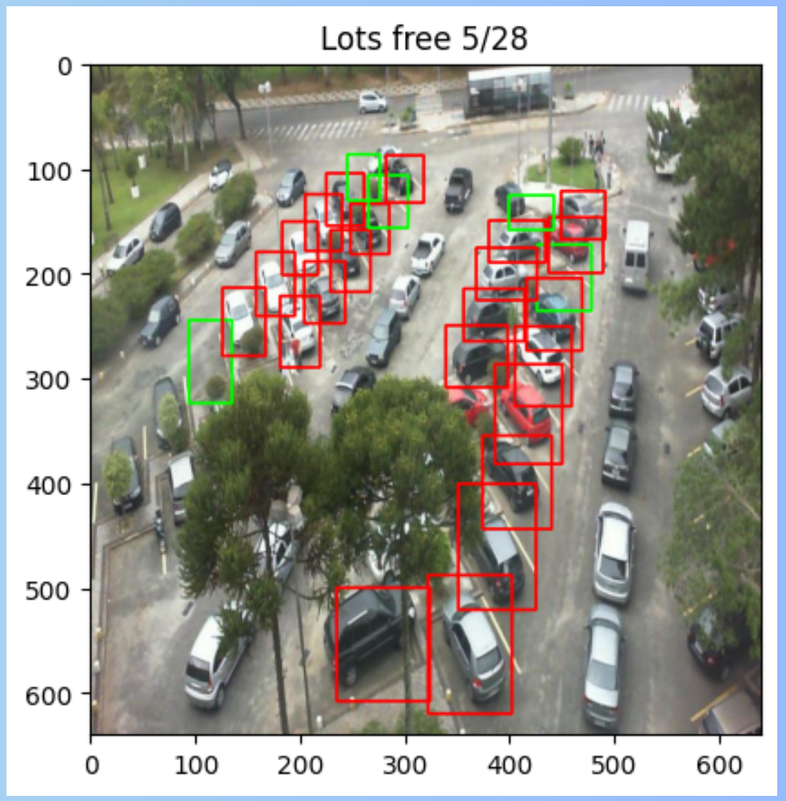
APPROCHOES TO FIND CARS

Use classificator car/no car on cropped images of parking area



APPROCHOES TO FIND CARS

Use classificator car/no car on cropped images of parking area



APPROCHOES TO FIND CARS

> Use classificator car/no car on cropped images of parking area

Pros

- More precise
- Works faster than the object detection algorithm when the number of objects is small

Cons

- works slower than the object detection algorithm when the number of objects is huge

CHOSEN APPROACH

1. Manually make a markup for every different location

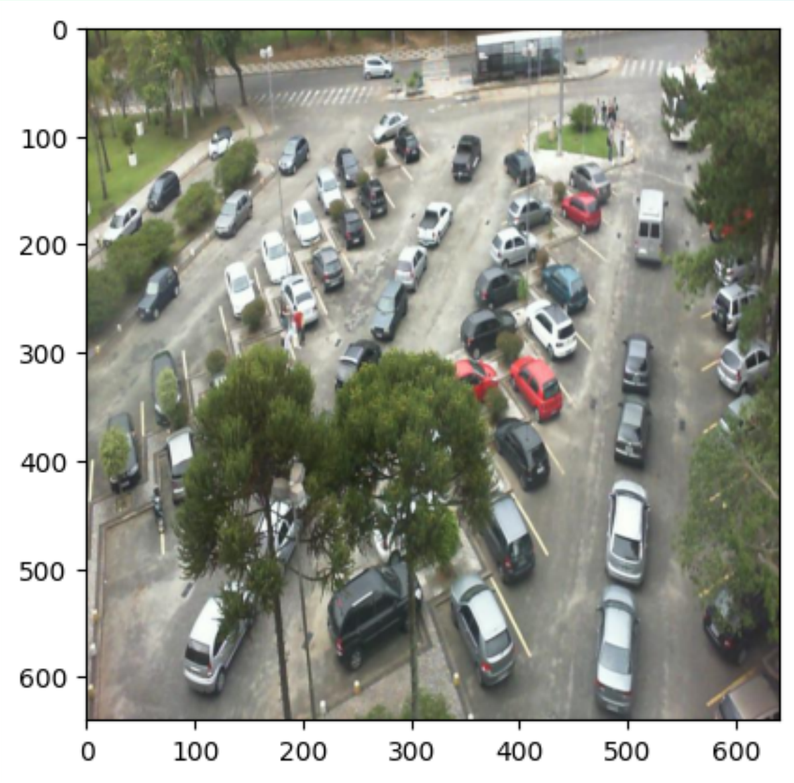
2. Extract parking spots image from the whole image

3. Run classification on every parking spots of every image

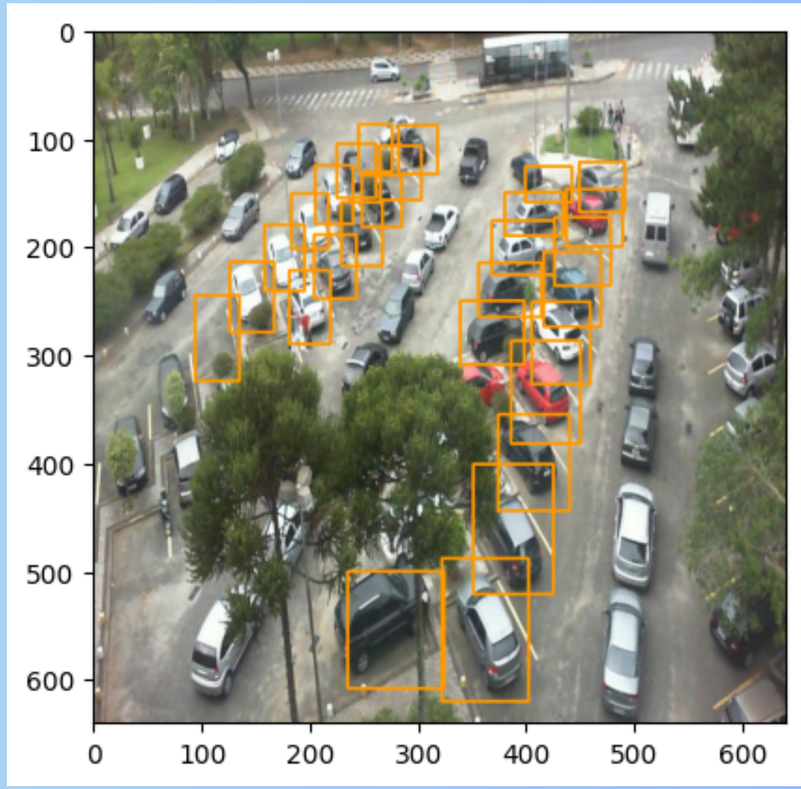
4. Count number of free parking spots from the return of classificator and visiualize them on image

STEP 1: MARK UP IMAGES

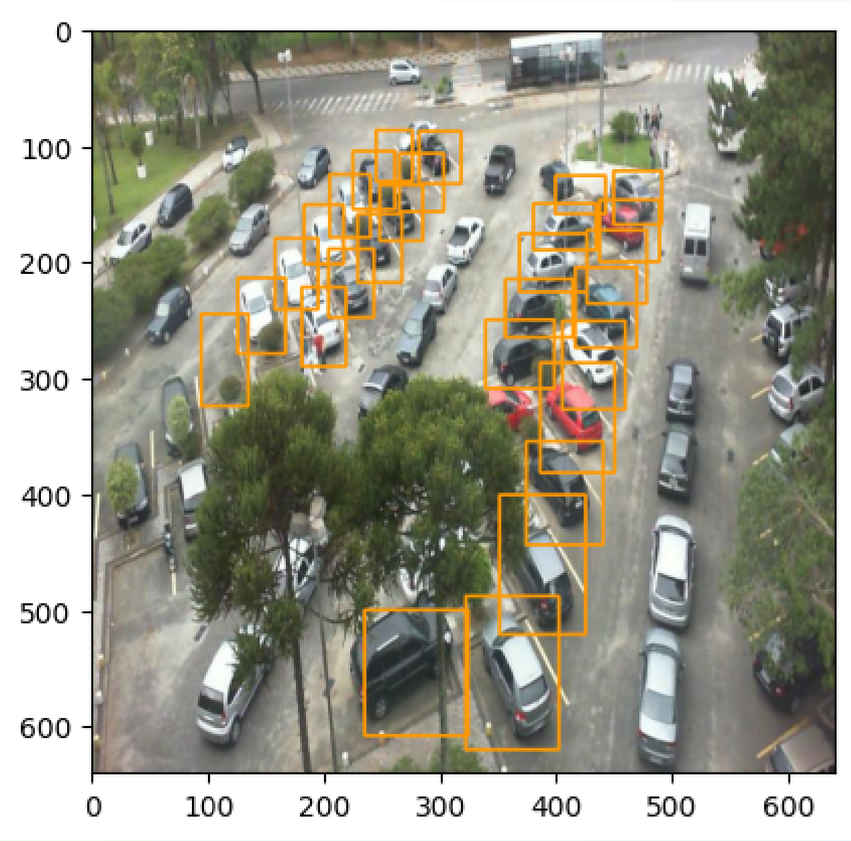
Markup: get (x ,y, h, w) coordinates of every spot of the location



polygon_extractor.py



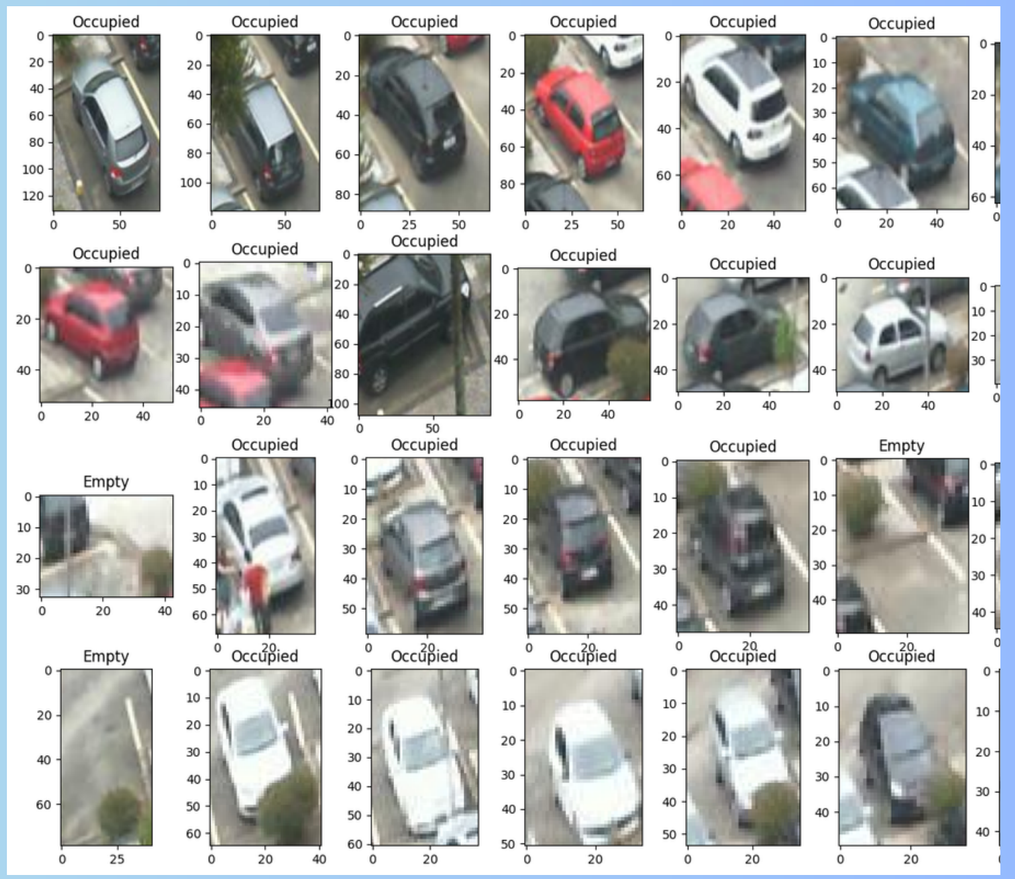
STEP 2: EXTRACT PARKING SPOTS



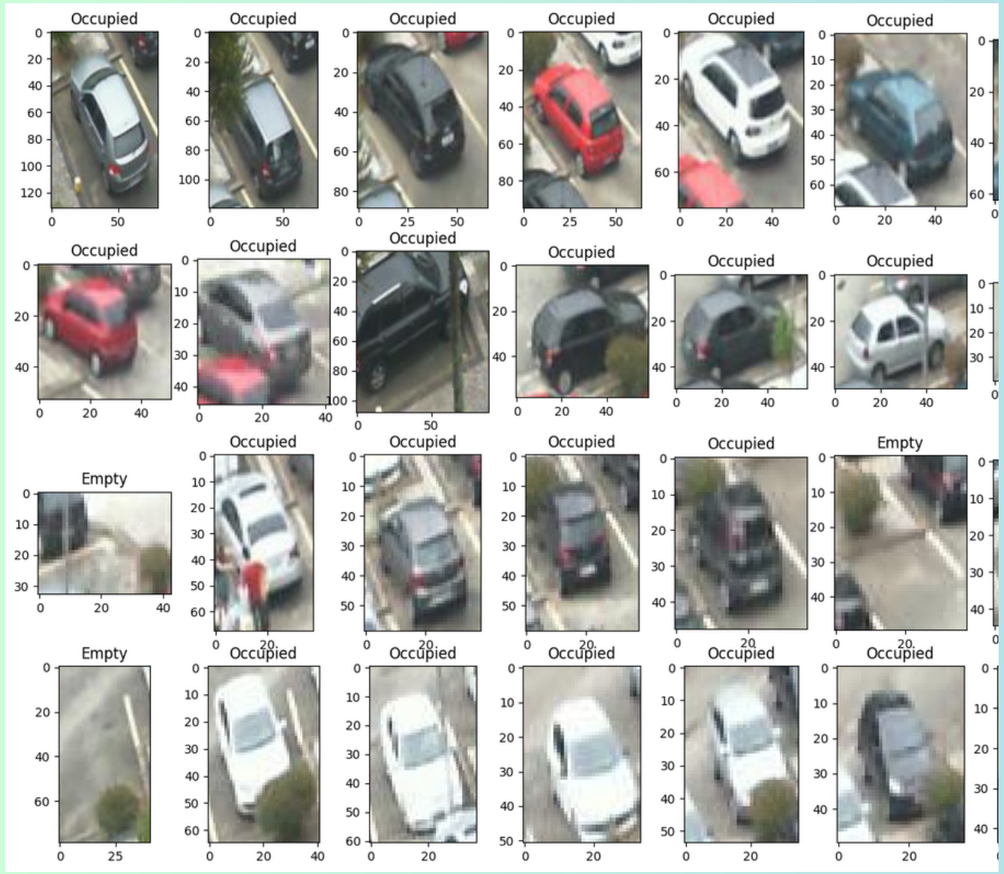
STEP 3: RUN CLASSIFIER

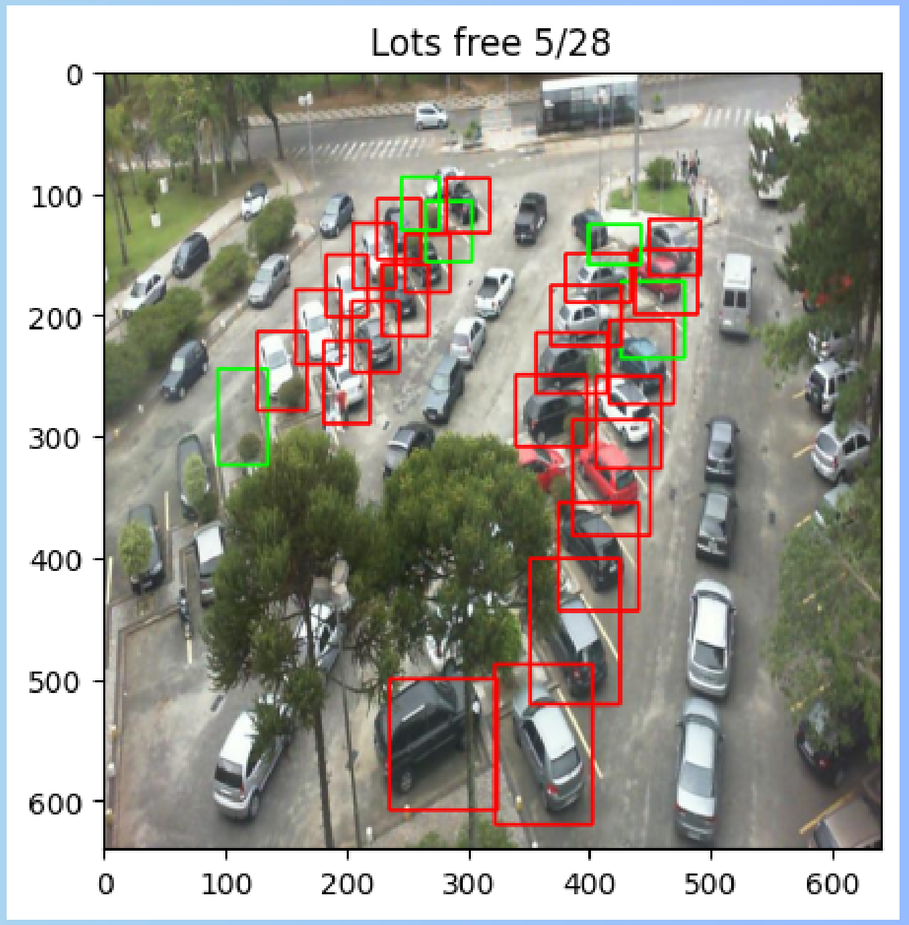
Used network: MobileNetv2

Execution time: 0.5 - 2s



STEP 4: GATHER RESULTS





MODEL SUMMARY

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 96, 96, 3)]       0

 tf.math.truediv (TFOpLambda  (None, 96, 96, 3)        0
 )

 tf.math.subtract (TFOpLambd  (None, 96, 96, 3)        0
 a)

 mobilenetv2_1.00_96 (Functi  (None, 3, 3, 1280)       2257984
 onal)

 global_average_pooling2d (G  (None, 1280)             0
 lobalAveragePooling2D)

 dropout (Dropout)           (None, 1280)              0

 dense (Dense)               (None, 1)                 1281

=================================================================
Total params: 2,259,265
Trainable params: 1,281
Non-trainable params: 2,257,984
_________________________________________________________________
```

METRICS

Train set:

MSE: 0.0479

Accuracy: 0.9980

Validation set

MSE: 0.0384

Accuracy: 0.9988

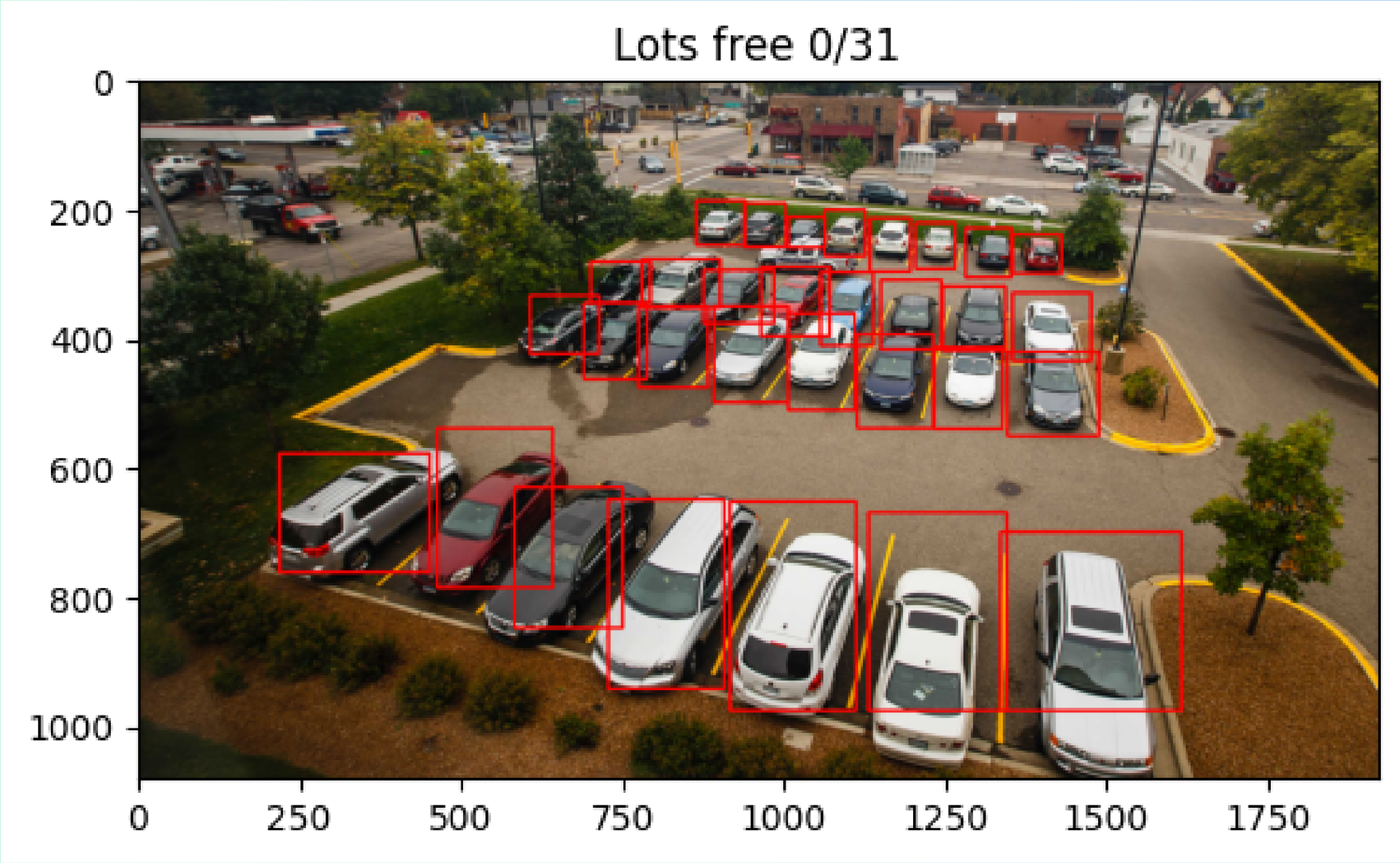
Test set:

MSE: 0.1245

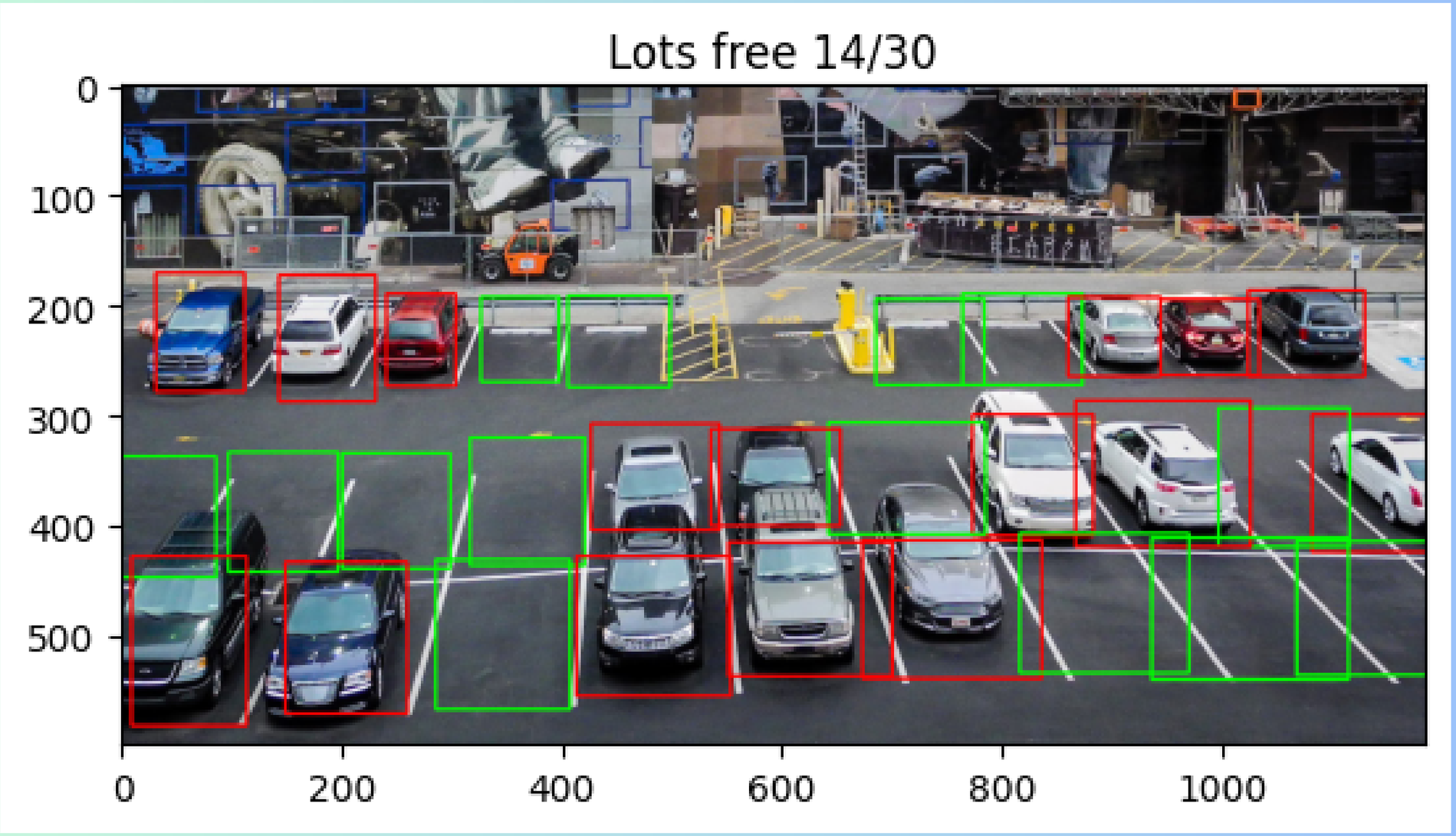
Accuracy: 0.9664

AUC: 0.9869

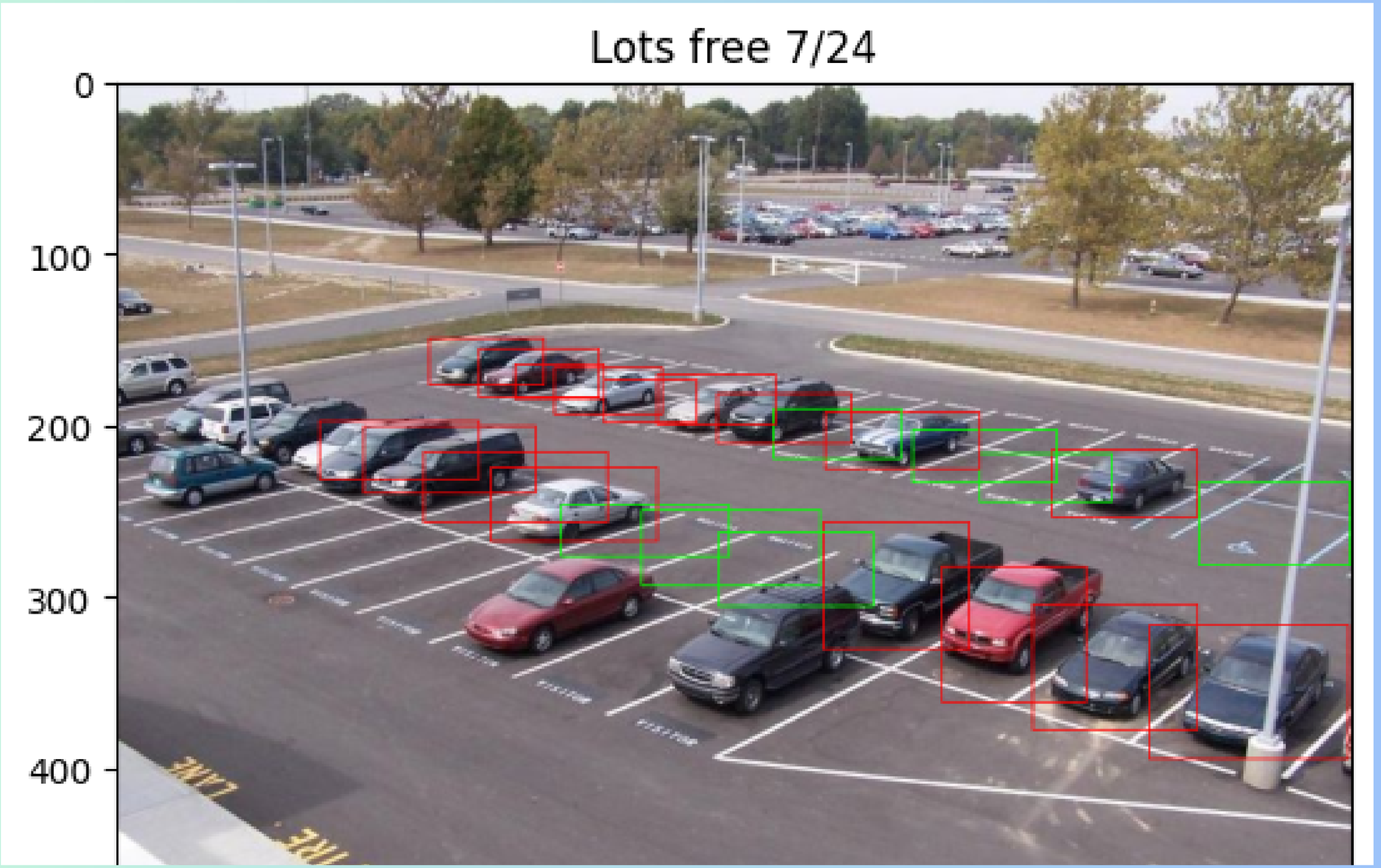
SOME CUSTOM TESTS



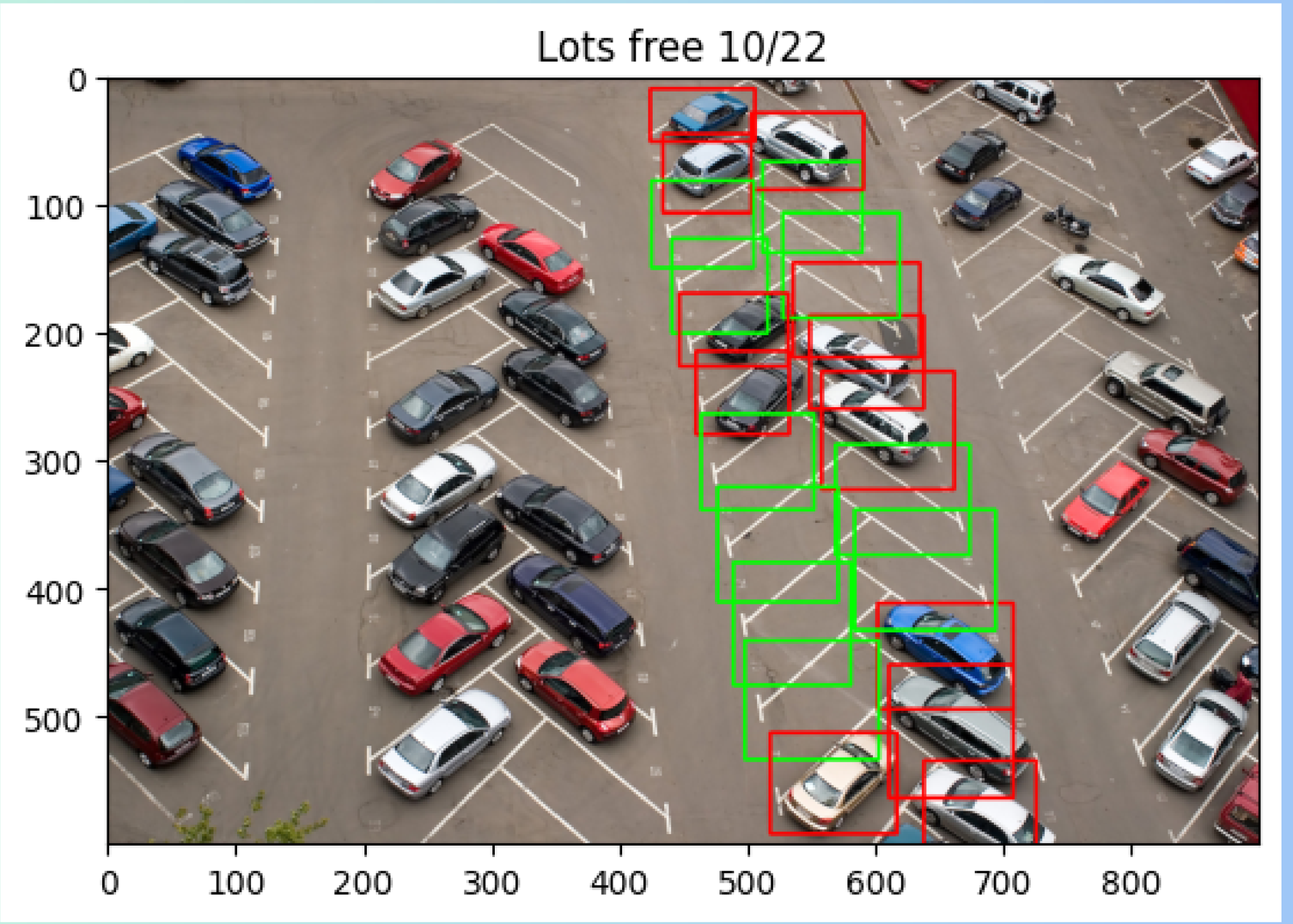
SOME CUSTOM TESTS



SOME CUSTOM TESTS

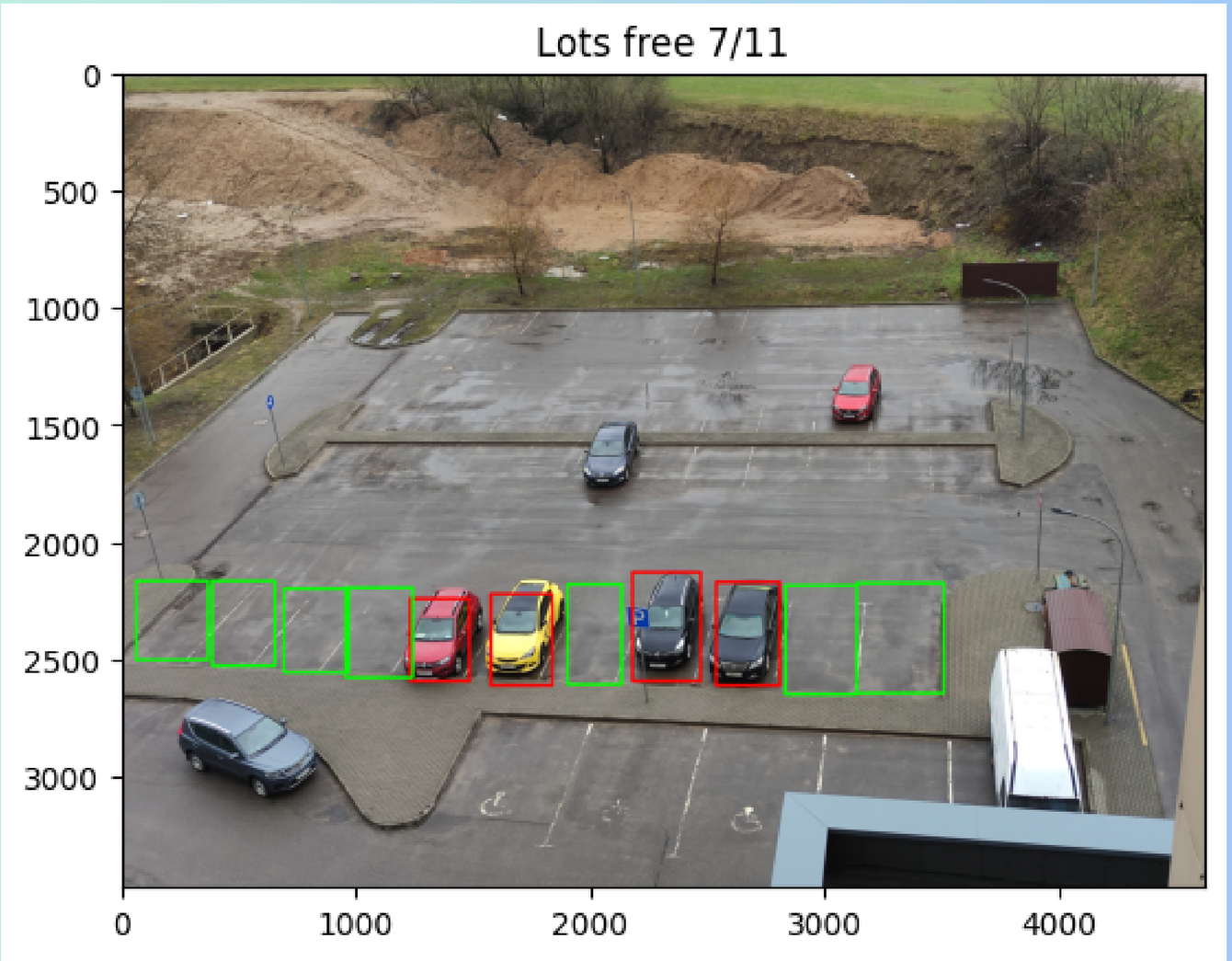


SOME CUSTOM TESTS



SOME CUSTOM TESTS

SOME CUSTOM TESTS



LIMITATIONS

- Can't give reasonbale results if camera is set to low
- Can't give predictions without manual markup
- Can't give reasonbale results if image has low resolution

THANK YOU FOR YOUR ATTENTION!