



License Plates Recognition

Sawsan Aljuhani

Idea:

- Detect the license plate and extract the text of the license plate number and text using custom model.
- This model could be linked with a real-time camera and monitor the traffic violations easily!

Prepare The Dataset

kaggle

Training

colab
TensorFlow

Object Detection

OpenCV

Text Recognition


Prepare the dataset:
























- 80% for Training
- 20% for Testing



- The Annotations for each image
- Includes the exact position of the object, and the class name

[Data](#) [Tasks \(1\)](#) [Code \(17\)](#) [Discussion](#) [Download \(213 MB\)](#) [New Notebook](#) ⋮

▼  annotations

 [Cars0.xml](#)
 Cars1.xml
 Cars10.xml
 Cars100.xml
 Cars101.xml
 Cars102.xml
 Cars103.xml
 Cars104.xml
 Cars105.xml
 Cars106.xml
 Cars107.xml
 Cars108.xml
 Cars109.xml
 Cars11.xml
 Cars110.xml
 Cars111.xml
 Cars112.xml
 Cars113.xml
 Cars114.xml
 Cars115.xml
 Cars116.xml
 Cars117.xml
 Cars118.xml

```
<annotation>
  <folder>images</folder>
  <filename>Cars0.png</filename>
  <size>
    <width>500</width>
    <height>268</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>licence</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <bndbox>
      <xmin>226</xmin>
      <ymin>125</ymin>
      <xmax>419</xmax>
      <ymax>173</ymax>
    </bndbox>
  </object>
</annotation>
```



Trining:

colab



TensorFlow

Set up the required files
and the configuration

Load Pre-trained model

Start the Training Using
Tensorflow

```
!python Tensorflow/models/research/object_detection/model_main_tf2.py --model_dir=Tensorflow/workspace/models/my_ssd_mob
```

```
INFO:tensorflow:Step 8700 per-step time 0.192s
I1215 23:14:30.973115 140268877838208 model_lib_v2.py:707] Step 8700 per-step time 0.192s
INFO:tensorflow:{'Loss/classification_loss': 0.12218677,
  'Loss/localization_loss': 0.075695276,
  'Loss/regularization_loss': 0.12859781,
  'Loss/total_loss': 0.32647985,
  'learning_rate': 0.07522382}
I1215 23:14:30.973505 140268877838208 model_lib_v2.py:708] {'Loss/classification_loss': 0.12218677,
  'Loss/localization_loss': 0.075695276,
  'Loss/regularization_loss': 0.12859781,
  'Loss/total_loss': 0.32647985,
  'learning_rate': 0.07522382}
INFO:tensorflow:Step 8800 per-step time 0.194s
```

Image from the training

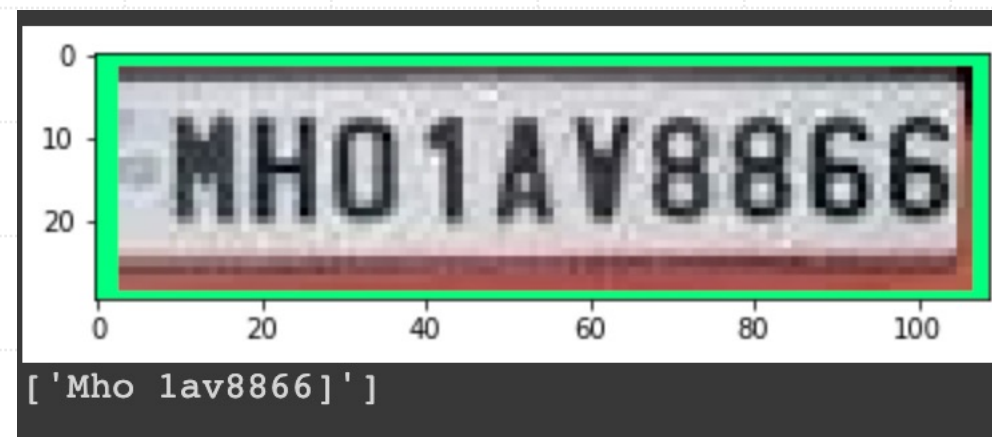
Object Detection:

- Using Tensorflow Object Detection



Detect the region of interest based in the trained model

Text Recognition:



Using EasyOCR to Extract the text

Challenges:

- YOLOv4 Model and Tesseract
- TensorFlow model was not accurate as YOLOv4 model





Thank You

Any Questions?