License Plate Detection using Detectron2, PyTorch, Colab, Roboflow and Tesseract

Objective

This notebook shows an end to end pipeline for detecting license plates from images using Detectron2, PyTorch, and Tesseract. It uses Google"c Colab for using GPUs, and Roboflow for generating Coco compatible image dataset. It also caches weights for a trained model for reuse during inference operations.

Tools and references

Here are references to the tools used:

Cars dataset: https://www.kaggle.com/andrewmvd/car-plate-detection

Detectron2: https://ai.facebook.com/tools/detectron2/

Tesseract: https://pypi.org/project/pytesseract/

Google Colab: https://research.google.com/colaboratory/faq.html

Roboflow: https://roboflow.com/

Coco data format: https://opencv.org/introduction-to-the-coco-dataset/#:~:text=COCO%20stores%20data%20in%20a,level%20description%20of%20the%20dataset

A related reference: https://towardsdatascience.com/object-detection-in-6-steps-using-detectron2-705b92575578

Image denoising: https://stackoverflow.com/questions/37745519/use-pytesseract-ocr-to-recognize-text-from-an-image

Computer Vision references: https://courses.opencv.org/dashboard

Links to my dataset:

Approach

This notebook uses Google Colab for access to GPUs.
The full pipeline uses the following steps:

1. Mount Colab
2. Install Detectron2 dependencies
3. Install Detectron2
4. Import packages
5. Register Dataset: The dataset needs to be registered with detectron2 to be used. The dataset format should be compatible with Coco format (see references). We used Roboflow to convert our data to Coco format (references)
6. Visualize training data. Visualizing data helps in ensuring we use the right dataset for training which can be computationally expensive.
7. Create CocoTrainer class.
8. Train the model.
9. Inspect training curves in tensorboard.
10. Define predictor and evaluator objects.
11. Save weights for the trained model.
12. List output files.
13. Assign proper weights to configuration object.
14. Denoise image.
15. Infer and visualize predictions.

Note

This notebook describes an end to end approach to detect license plates from car images. I will update these notes with approaches which enhance the accuracy and execution time as I get additional information.

Summary

We are usng detectron2 to localize the license plate bounding box of a car. This localized license plate image is fed to an OCR tool like Tesseract for licens plate detection. We could improve the accuracy of detection by denoising the image before feeding it to the OCR tool. Another approach would be to use third party APIs for OCR function like:

https://platerecognizer.com/alpr-results

1. Mount colab

```
# Note:  Change runtime type to a GPU
from google.colab import drive
drive.mount("/content/drive", force_remount=True)

print('done')
```

```
Mounted at /content/drive
done
```

2. **Install Detectron2 Dependencies**

```
#Note:  use cu101 because colab has CUDA 10.1

!pip install -U torch==1.5 torchvision==0.6 -f https://download.pytorch.org/whl/cu101/torch_stable.html
!pip install cython pyyaml==5.1
!pip install -U 'git+https://github.com/cocodataset/cocoapi.git#subdirectory=PythonAPI'
import torch, torchvision
print(torch.__version__, torch.cuda.is_available())
!gcc --version
import random
#------------------------------
# run parameters
is_train = 0 # set to 1 for training model for a new dataset, or 0 to use weights from a previous training run
model_path = '/content/drive/MyDrive/content2/datasetsAndJNs/text_detection/data/cached_model'
max_iter = 1000
score_threshold_test = 0.75
#-----------------------------------------------
train_jsons  =  "/content/drive/MyDrive/content2/datasetsAndJNs/cars_kaggle/data_coco/train/_annotations.coco.json"
train_images =  "/content/drive/MyDrive/content2/datasetsAndJNs/cars_kaggle/data_coco/train"

valid_jsons  =  "/content/drive/MyDrive/content2/datasetsAndJNs/cars_kaggle/data_coco/valid/_annotations.coco.json"
valid_images =  "/content/drive/MyDrive/content2/datasetsAndJNs/cars_kaggle/data_coco/valid"

test_jsons  =  "/content/drive/MyDrive/content2/datasetsAndJNs/cars_kaggle/data_coco/test/_annotations.coco.json"
test_images =  "/content/drive/MyDrive/content2/datasetsAndJNs/cars_kaggle/data_coco/test"

test_lp_jsons  =  "/content/drive/MyDrive/content2/datasetsAndJNs/cars_kaggle/data_coco/test_lp/_annotations.coco.json"
test_lp_images =  "/content/drive/MyDrive/content2/datasetsAndJNs/cars_kaggle/data_coco/test_lp"
##-------------------------------------------------------
# train_jsons  =  "path to your train json file"
# train_images =  "path to your train images'

# valid_jsons  =  "path to your valid json file"
# valid_images =  "path to your valid images"

# test_jsons  =  "path to your test json file"
# test_images =  "path to your test images"

# test_lp_jsons  =  "path to your test_lp json file"
# test_lp_images =  "path to your test_lp images"
#-------------------------------------------------------

dsv = str(random.randint(0,1000000))  ## dataset version

print('done')
```

```
  Installing collected packages: pyyaml
    Attempting uninstall: pyyaml
      Found existing installation: PyYAML 6.0
      Uninstalling PyYAML-6.0:
        Successfully uninstalled PyYAML-6.0
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviou
fastai 2.7.10 requires torch<1.14,>=1.7, but you have torch 1.5.0+cu101 which is incompatible.
fastai 2.7.10 requires torchvision>=0.8.2, but you have torchvision 0.6.0+cu101 which is incompatible.
dask 2022.2.1 requires pyyaml>=5.3.1, but you have pyyaml 5.1 which is incompatible.
Successfully installed pyyaml-5.1
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting git+https://github.com/cocodataset/cocoapi.git#subdirectory=PythonAPI
  Cloning https://github.com/cocodataset/cocoapi.git to /tmp/pip-req-build-ayc2iz5e
  Running command git clone -q https://github.com/cocodataset/cocoapi.git /tmp/pip-req-build-ayc2iz5e
Requirement already satisfied: setuptools>=18.0 in /usr/local/lib/python3.8/dist-packages (from pycocotools==2.0) (57.4.
Requirement already satisfied: cython>=0.27.3 in /usr/local/lib/python3.8/dist-packages (from pycocotools==2.0) (0.29.32
Requirement already satisfied: matplotlib>=2.1.0 in /usr/local/lib/python3.8/dist-packages (from pycocotools==2.0) (3.2.2
Requirement already satisfied: numpy>=1.11 in /usr/local/lib/python3.8/dist-packages (from matplotlib>=2.1.0->pycocotools
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib>=2.1.0->pycoc
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.8/dist-packages (from m
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-packages (from matplotlib>=2.1.0->pycocotool
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib>=2.1.0->py
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-packages (from python-dateutil>=2.1->matplotlib>
Building wheels for collected packages: pycocotools
  Building wheel for pycocotools (setup.py) ... done
  Created wheel for pycocotools: filename=pycocotools-2.0-cp38-cp38-linux_x86_64.whl size=304485 sha256=22869d48611910a01
  Stored in directory: /tmp/pip-ephem-wheel-cache-i9sazq7y/wheels/56/da/49/cb71a7c450b59588934077f431100c05fbde50646ee84a
Successfully built pycocotools
Installing collected packages: pycocotools
  Attempting uninstall: pycocotools
    Found existing installation: pycocotools 2.0.6
    Uninstalling pycocotools-2.0.6:
      Successfully uninstalled pycocotools-2.0.6
Successfully installed pycocotools-2.0
1.5.0+cu101 True
gcc (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

3. install Detectron2

```
!pip install detectron2==0.1.3 -f https://dl.fbaipublicfiles.com/detectron2/wheels/cu101/torch1.5/index.html

print('done')
```

```
Requirement already satisfied: pydot in /usr/local/lib/python3.8/dist-packages (from detectron2==0.1.3) (1.3.0)
Requirement already satisfied: termcolor>=1.1 in /usr/local/lib/python3.8/dist-packages (from detectron2==0.1.3) (2.1.1)
Collecting mock
  Downloading mock-4.0.3-py3-none-any.whl (28 kB)
Requirement already satisfied: tensorboard in /usr/local/lib/python3.8/dist-packages (from detectron2==0.1.3) (2.9.1)
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.8/dist-packages (from detectron2==0.1.3) (1.5.0)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.8/dist-packages (from detectron2==0.1.3) (3.2.2)
Requirement already satisfied: tabulate in /usr/local/lib/python3.8/dist-packages (from detectron2==0.1.3) (0.8.10)
Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages (from fvcore>=0.1.1->detectron2==0.1.3) (1
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.8/dist-packages (from fvcore>=0.1.1->detectron2==0.1
Collecting iopath>=0.1.7
  Downloading iopath-0.1.10.tar.gz (42 kB)
     |████████████████████████████████| 42 kB 918 kB/s
Requirement already satisfied: typing_extensions in /usr/local/lib/python3.8/dist-packages (from iopath>=0.1.7->fvcore>=0
Collecting portalocker
  Downloading portalocker-2.6.0-py2.py3-none-any.whl (15 kB)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib->detectron
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.8/dist-packages (from m
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-packages (from matplotlib->detectron2==0.1.3
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib->detectron2==
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-packages (from python-dateutil>=2.1->matplotlib-
```

```
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.8/dist-packages (from google-auth<3,>=1.6.3->tens
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.8/dist-packages (from google-auth<3,>=1.6
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.8/dist-packages (from google-auth-oauth
Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.8/dist-packages (from markdown>=2.6.8->t
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.8/dist-packages (from importlib-metadata>=4.4->markdow
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.8/dist-packages (from pyasn1-modules>=0.2.1
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-packages (from requests<3,>=2.21.0->tensorbo
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.8/dist-packages (from re
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.8/dist-packages (from requests<3,>=2.21.0->te
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.8/dist-packages (from requests<3,>=2.21.0->ten
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.8/dist-packages (from requests-oauthlib>=0.7.0->
Building wheels for collected packages: fvcore, iopath
  Building wheel for fvcore (setup.py) ... done
  Created wheel for fvcore: filename=fvcore-0.1.5.post20221221-py3-none-any.whl size=61431 sha256=d1b3f4ca50139db63ce12e0
  Stored in directory: /root/.cache/pip/wheels/b8/79/07/c0e9367f5b5ea325e246bd73651e8af175fabbef943043b1cc
  Building wheel for iopath (setup.py) ... done
  Created wheel for iopath: filename=iopath-0.1.10-py3-none-any.whl size=31547 sha256=87f9614e141708f1bfdf0071ca15545a674
  Stored in directory: /root/.cache/pip/wheels/89/3e/24/0f349c0b2eeb6965903035f3b00dbb5c9bea437b4a2f18d82c
Successfully built fvcore iopath
Installing collected packages: portalocker, yacs, iopath, mock, fvcore, detectron2
```

4. imports

```
import detectron2
from detectron2.utils.logger import setup_logger
setup_logger()

# import some common libraries
import numpy as np
import os
import cv2
import random
from google.colab.patches import cv2_imshow

# import some common detectron2 utilities
from detectron2 import model_zoo
from detectron2.engine import DefaultPredictor
from detectron2.config import get_cfg
from detectron2.utils.visualizer import Visualizer
from detectron2.data import MetadataCatalog
from detectron2.data.catalog import DatasetCatalog

print('done')
```

```
done
```

5. register datasets

```
from detectron2.data.datasets import register_coco_instances
##------------------------------------------------------------
# !ls /content/drive/MyDrive/content2/datasetsAndJNs/cars_kaggle/data_coco/train/_annotations.coco.json

# !ls /content/drive/MyDrive/content2/datasetsAndJNs/cars_kaggle/data_coco/train

# !ls /content/drive/MyDrive/content2/datasetsAndJNs/cars_kaggle/data_coco/valid/_annotations.coco.json

# !ls /content/drive/MyDrive/content2/datasetsAndJNs/cars_kaggle/data_coco/valid

# !ls /content/drive/MyDrive/content2/datasetsAndJNs/cars_kaggle/data_coco/test/_annotations.coco.json

# !ls /content/drive/MyDrive/content2/datasetsAndJNs/cars_kaggle/data_coco/test

# !ls /content/drive/MyDrive/content2/datasetsAndJNs/cars_kaggle/data_coco/test_lp/_annotations.coco.json

# !ls /content/drive/MyDrive/content2/datasetsAndJNs/cars_kaggle/data_coco/test_lp

register_coco_instances("my_dataset_train" + dsv, {}, train_jsons, train_images)

register_coco_instances("my_dataset_valid" + dsv, {}, valid_jsons, valid_images)

register_coco_instances("my_dataset_test" + dsv, {}, test_jsons, test_images)

register_coco_instances("my_dataset_test_lp" + dsv, {}, test_jsons, test_lp_images)
```

```
print('done')
```

```
done
```

6. **visualize training data**

```
my_dataset_train_metadata = MetadataCatalog.get("my_dataset_train" + dsv)
dataset_dicts = DatasetCatalog.get("my_dataset_train" + dsv)

import random
from detectron2.utils.visualizer import Visualizer

for d in random.sample(dataset_dicts, 3):
    fname = d["file_name"]
    img = cv2.imread(fname)
    visualizer = Visualizer(img[:, :, ::-1], metadata=my_dataset_train_metadata, scale=0.5)
    vis = visualizer.draw_dataset_dict(d)
    cv2_imshow(vis.get_image()[:, :, ::-1])
```

```
[12/28 06:36:28 d2.data.datasets.coco]: Loading /content/drive/MyDrive/content2/
WARNING [12/28 06:36:28 d2.data.datasets.coco]:
```

7. train Custom Detectron2 Detector

```
# We are importing our own Trainer Module here to use the COCO validation evaluation during training. Otherwise no validation eva

from detectron2.engine import DefaultTrainer
from detectron2.evaluation import COCOEvaluator

class CocoTrainer(DefaultTrainer):

  @classmethod
  def build_evaluator(cls, cfg, dataset_name, output_folder=None):
    coco_eval = '/content/drive/MyDrive/content2/datasetsAndJNs/cars_kaggle/data_coco'
    if output_folder is None:
        os.makedirs(coco_eval, exist_ok=True)
        output_folder = coco_eval

    return COCOEvaluator(dataset_name, cfg, False, output_folder)

print('done')
```

```
done
```

8. train

```
from detectron2.engine import DefaultTrainer

cfg = get_cfg()
cfg.merge_from_file(model_zoo.get_config_file("COCO-Detection/faster_rcnn_X_101_32x8d_FPN_3x.yaml"))

cfg.DATASETS.TRAIN = ("my_dataset_train" + dsv,)
cfg.DATASETS.TEST = ("my_dataset_valid" + dsv,) ## datasets.test should be valid dataset

cfg.DATALOADER.NUM_WORKERS = 2

cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url("COCO-Detection/faster_rcnn_X_101_32x8d_FPN_3x.yaml")  # Let training initialize

cfg.SOLVER.IMS_PER_BATCH = 2  # This is the real "batch size" commonly known to deep learning people
cfg.SOLVER.BASE_LR = 0.00025  # pick a good LR
cfg.SOLVER.MAX_ITER = max_iter    # 300 iterations seems good enough for this toy dataset; you will need to train longer for a pr
cfg.SOLVER.STEPS = []        # do not decay learning rate
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 128   # The "RoIHead batch size". 128 is faster, and good enough for this toy dataset
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 2  # only has one class (ballon). (see https://detectron2.readthedocs.io/tutorials/datasets.htm
# NOTE: this config means the number of classes, but a few popular unofficial tutorials incorrect uses num_classes+1 here.

os.makedirs(cfg.OUTPUT_DIR, exist_ok=True)
trainer = DefaultTrainer(cfg)
trainer.resume_or_load(resume=False)
if(is_train==1):
  trainer.train()
else:
  trainer = torch.load(model_path + '/model_final.pth')

print('done')
```

```
          (2): ROIAlign(output_size=(7, 7), spatial_scale=0.0625, sampling_ratio=0, aligned=True)
          (3): ROIAlign(output_size=(7, 7), spatial_scale=0.03125, sampling_ratio=0, aligned=True)
        )
      )
      (box_head): FastRCNNConvFCHead(
        (fc1): Linear(in_features=12544, out_features=1024, bias=True)
        (fc2): Linear(in_features=1024, out_features=1024, bias=True)
      )
      (box_predictor): FastRCNNOutputLayers(
        (cls_score): Linear(in_features=1024, out_features=3, bias=True)
        (bbox_pred): Linear(in_features=1024, out_features=8, bias=True)
      )
    )
  )
WARNING [12/28 06:36:39 d2.data.datasets.coco]:
Category ids in annotations are not in [1, #categories]! We'll apply a mapping for you.

[12/28 06:36:39 d2.data.datasets.coco]: Loaded 215 images in COCO format from /content/drive/MyDrive/content2/datasetsAnd
[12/28 06:36:39 d2.data.build]: Removed 0 images with no usable annotations. 215 images left.
[12/28 06:36:39 d2.data.build]: Distribution of instances among all 2 categories:
|  category  | #instances   |  category  | #instances   |
|:----------:|:-------------|:----------:|:-------------|
|  letters   | 0            |  licence   | 242          |
|            |              |            |              |
|   total    | 242          |            |              |
[12/28 06:36:39 d2.data.common]: Serializing 215 elements to byte tensors and concatenating them all ...
[12/28 06:36:39 d2.data.common]: Serialized dataset takes 0.07 MiB
[12/28 06:36:39 d2.data.detection_utils]: TransformGens used in training: [ResizeShortestEdge(short_edge_length=(640, 672
[12/28 06:36:39 d2.data.build]: Using training sampler TrainingSampler
model_final_68b088.pkl: 421MB [00:07, 57.5MB/s]
WARNING:fvcore.common.checkpoint:Skip loading parameter 'roi_heads.box_predictor.cls_score.weight' to the model due to in
WARNING:fvcore.common.checkpoint:Skip loading parameter 'roi_heads.box_predictor.cls_score.bias' to the model due to incom
WARNING:fvcore.common.checkpoint:Skip loading parameter 'roi_heads.box_predictor.bbox_pred.weight' to the model due to in
WARNING:fvcore.common.checkpoint:Skip loading parameter 'roi_heads.box_predictor.bbox_pred.bias' to the model due to incom
WARNING:fvcore.common.checkpoint:Some model parameters or buffers are not found in the checkpoint:
roi_heads.box_predictor.bbox_pred.{bias, weight}
```

9. look at training curves in tensorboard

```
%load_ext tensorboard
%tensorboard --logdir output
```

TensorBoard INACTIVE

```
# This is formatted as code
```

10. **Define predictor and evaluator objects**

```
from detectron2.data import DatasetCatalog, MetadataCatalog, build_detection_test_loader
from detectron2.evaluation import COCOEvaluator, inference_on_dataset

weights_path = cfg.OUTPUT_DIR
if(is_train!=1):
  weights_path = model_path

cfg.MODEL.WEIGHTS = os.path.join(weights_path, "model_final.pth")

cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = score_threshold_test
predictor = DefaultPredictor(cfg)
evaluator = COCOEvaluator("my_dataset_test" + dsv, cfg, False, output_dir="./output/")
val_loader = build_detection_test_loader(cfg, "my_dataset_test" + dsv)
if(is_train==1):
  inference_on_dataset(trainer.model, val_loader, evaluator)

print('done')
```

```
WARNING [12/28 06:37:01 d2.data.datasets.coco]:
Category ids in annotations are not in [1, #categories]! We'll apply a mapping for you.

[12/28 06:37:01 d2.data.datasets.coco]: Loaded 29 images in COCO format from /content/drive/MyDrive/content2/datasetsAndJNs/
[12/28 06:37:01 d2.data.build]: Distribution of instances among all 2 categories:
|  category  | #instances   |  category  | #instances   |
|:----------:|:-------------|:----------:|:-------------|
|  letters   | 0            |  licence   | 31           |
|            |              |            |              |
|   total    | 31           |            |              |
[12/28 06:37:01 d2.data.common]: Serializing 29 elements to byte tensors and concatenating them all ...
[12/28 06:37:01 d2.data.common]: Serialized dataset takes 0.01 MiB
done
```

11, **save weights**

```
from detectron2.checkpoint import DetectionCheckpointer, Checkpointer

if (is_train==1):
  checkpointer = DetectionCheckpointer(trainer.model, save_dir=model_path)
  checkpointer.save("model_final")
#----------------------
print('done')
```

```
done
```

12. **List output files**

```
%ls ./output/
```

```
events.out.tfevents.1672209402.d164c191e916.304.0  metrics.json
```

13. **Assign proper weights to configuration object**

```
cfg.DATASETS.TEST = ("my_dataset_test" + dsv, )

test_metadata = MetadataCatalog.get("my_dataset_test" + dsv)

print('done')
```

```
done
```

14. **Denoise input image**

```
def denoise(image):
  gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
  blur = cv2.GaussianBlur(gray, (3,3), 0)
  thresh = cv2.threshold(blur, 150, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)[1]

  # Morph open to remove noise and invert image
  kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (3,3))
  opening = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, kernel, iterations=1)
  invert = 255 - opening
  return invert
```

15. **infer and visualize perdctions**

```
from detectron2.utils.visualizer import ColorMode
import glob

#-------------
!sudo apt-get install tesseract-ocr
!pip install pytesseract==0.3.9
import pytesseract
from pytesseract import Output
from PIL import Image
import cv2
#------------
bboxes2 = []
ct_cars_detected = 0
ct_images = 0
cars_undetected = []
cars_detected = []
cars_lp_detected = []
cars_all = []
for imageName in glob.glob(test_lp_images + '/*jpg'):
  ct_images = ct_images + 1
  cars_all.append(imageName)
  im = cv2.imread(imageName)
  outputs = predictor(im)

  output_pred_boxes = outputs["instances"].pred_boxes

  for i in output_pred_boxes.__iter__():
    bb = i.cpu().numpy()
    flag = np.any(bb)
    if(flag):
        bboxes2.append((imageName, bb))
        print('bounding box', bb)
        ct_cars_detected = ct_cars_detected + 1
        bboxes2.append((imageName, bb))
        #-------------------------------------
        y1 = int(bb[0])
        x1 = int(bb[1])
        y2 = int(bb[2])
        x2 = int(bb[3])

        print('x1,y1,x2, y2', x1, y1, x2, y2)
        license_plate = im[x1:x2, y1:y2]
        print('image')
        cv2_imshow(im)
        print('license_plate')
        cv2_imshow(license_plate)
        im_denoise = denoise(im)

        cv2.waitKey(0)
        #------------------
```

```
            lpn = pytesseract.image_to_string(license_plate)
            print('license plate number: ', lpn)

            # lpn_denoise = pytesseract.image_to_string(im_denoise)
            # print('license_plate_number_denoise: ', lpn_denoise)

    print('done')
```

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libnvidia-common-460
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  tesseract-ocr-eng tesseract-ocr-osd
The following NEW packages will be installed:
  tesseract-ocr tesseract-ocr-eng tesseract-ocr-osd
0 upgraded, 3 newly installed, 0 to remove and 20 not upgraded.
Need to get 4,795 kB of archives.
After this operation, 15.8 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/universe amd64 tesseract-ocr-eng all 4.00~git24-0e00fe6-1.2 [1,588
Get:2 http://archive.ubuntu.com/ubuntu bionic/universe amd64 tesseract-ocr-osd all 4.00~git24-0e00fe6-1.2 [2,989
Get:3 http://archive.ubuntu.com/ubuntu bionic/universe amd64 tesseract-ocr amd64 4.00~git2288-10f4998a-2 [218 kB]
Fetched 4,795 kB in 1s (3,700 kB/s)
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based frontend cannot be used. at /usr/share/
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Selecting previously unselected package tesseract-ocr-eng.
(Reading database ... 124016 files and directories currently installed.)
Preparing to unpack .../tesseract-ocr-eng_4.00~git24-0e00fe6-1.2_all.deb ...
Unpacking tesseract-ocr-eng (4.00~git24-0e00fe6-1.2) ...
Selecting previously unselected package tesseract-ocr-osd.
Preparing to unpack .../tesseract-ocr-osd_4.00~git24-0e00fe6-1.2_all.deb ...
Unpacking tesseract-ocr-osd (4.00~git24-0e00fe6-1.2) ...
Selecting previously unselected package tesseract-ocr.
Preparing to unpack .../tesseract-ocr_4.00~git2288-10f4998a-2_amd64.deb ...
Unpacking tesseract-ocr (4.00~git2288-10f4998a-2) ...
Setting up tesseract-ocr-osd (4.00~git24-0e00fe6-1.2) ...
Setting up tesseract-ocr-eng (4.00~git24-0e00fe6-1.2) ...
Setting up tesseract-ocr (4.00~git2288-10f4998a-2) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting pytesseract==0.3.9
  Downloading pytesseract-0.3.9-py2.py3-none-any.whl (14 kB)
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.8/dist-packages (from pytesseract==0.3.9
Collecting Pillow>=8.0.0
  Downloading Pillow-9.3.0-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.2 MB)
     |████████████████████████████████| 3.2 MB 4.8 MB/s
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.8/dist-packages (from packaging
Installing collected packages: Pillow, pytesseract
  Attempting uninstall: Pillow
    Found existing installation: Pillow 7.1.2
    Uninstalling Pillow-7.1.2:
      Successfully uninstalled Pillow-7.1.2
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This b
fastai 2.7.10 requires torch<1.14,>=1.7, but you have torch 1.5.0+cu101 which is incompatible.
fastai 2.7.10 requires torchvision>=0.8.2, but you have torchvision 0.6.0+cu101 which is incompatible.
Successfully installed Pillow-9.3.0 pytesseract-0.3.9
WARNING: The following packages were previously imported in this runtime:
  [PIL]
You must restart the runtime in order to use newly installed versions.
```

RESTART RUNTIME

```
bounding box [177.58813 232.0496  362.42038 330.4094 ]
x1,y1,x2, y2 232 177 330 362
image
```

```
license_plate
```


```
license plate number:  Maly ATE

bounding box [222.1492  449.62222 400.8051  491.848  ]
x1,y1,x2, y2 449 222 491 400
image
```


```
license_plate
```


```
license plate number:
bounding box [207.05205 280.16153 418.82834 367.1024 ]
x1,y1,x2, y2 280 207 367 418
image
```