

Capstone project - CV2 - Cars Object detection

Summary of problem statement, data and findings

The requirement of this project is to classify and detect cars in the image dataset provided.

```
# Initializing the random number generator
import random
random.seed(0)
# Ignoring the warnings
import warnings
warnings.filterwarnings("ignore")

# Mounting the google drive to load the data file
from google.colab import drive
drive.mount('/content/drive/')

import os
path = '/content/drive/My Drive/capstone/Dataset'
capstone_path = '/content/drive/My Drive/capstone'
os.chdir(path)

↳ Mounted at /content/drive/
```

```
!pip uninstall opencv-python-headless
```

```
Found existing installation: opencv-python-headless 4.1.2.30
Uninstalling opencv-python-headless-4.1.2.30:
  Would remove:
    /usr/local/lib/python3.7/dist-packages/cv2/*
    /usr/local/lib/python3.7/dist-packages/opencv_python_headless-4.1.2.30.dist-info,
  Would not remove (might be manually added):
    /usr/local/lib/python3.7/dist-packages/cv2/.libs/libQtCore-bbdab771.so.4.8.7
    /usr/local/lib/python3.7/dist-packages/cv2/.libs/libQtGui-903938cd.so.4.8.7
    /usr/local/lib/python3.7/dist-packages/cv2/.libs/libQtTest-1183da5d.so.4.8.7
Proceed (y/n)? y
Successfully uninstalled opencv-python-headless-4.1.2.30
```

```
!pip install opencv-python-headless==4.1.2.30
```

```
Collecting opencv-python-headless==4.1.2.30
  Using cached opencv_python_headless-4.1.2.30-cp37-cp37m-manylinux1_x86_64.whl (21.8MB)
Requirement already satisfied: numpy>=1.14.5 in /usr/local/lib/python3.7/dist-packages
Installing collected packages: opencv-python-headless
Successfully installed opencv-python-headless-4.1.2.30
```

```
!pip install imgaug
```

```
Requirement already satisfied: imgaug in /usr/local/lib/python3.7/dist-packages (0.2
Requirement already satisfied: Shapely in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: imageio in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: opencv-python in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: Pillow in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from im
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from im
Requirement already satisfied: scikit-image>=0.11.0 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: PyWavelets>=1.1.1 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: networkx>=2.0 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: pyparsing!=2.0.4,!>=2.1.2,!>=2.1.6,>=2.0.1 in /usr/loc
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages
```

```
!pip install imagecorruptions
```

```
Collecting imagecorruptions
  Downloading imagecorruptions-1.1.2-py3-none-any.whl (2.1 MB)
    ██████████ | 2.1 MB 8.4 MB/s
Requirement already satisfied: Pillow>=5.4.1 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: opencv-python>=3.4.5 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: scipy>=1.2.1 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: numpy>=1.16 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: scikit-image>=0.15 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: networkx>=2.0 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: PyWavelets>=1.1.1 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: matplotlib!=3.0.0,>=2.0.0 in /usr/local/lib/python3.7/
Requirement already satisfied: imageio>=2.3.0 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: pyparsing!=2.0.4,!>=2.1.2,!>=2.1.6,>=2.0.1 in /usr/loc
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (fr
Installing collected packages: imagecorruptions
Successfully installed imagecorruptions-1.1.2
```

```
import numpy as np
import scipy as sp
import matplotlib as mpl
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
%matplotlib inline
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import VGG16
from tensorflow.keras import layers
```

```
from tensorflow.keras import models
from tensorflow.keras import regularizers, optimizers
from keras.models import Sequential
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Input
from tensorflow.keras.models import Model
from keras.layers import Conv2D, MaxPooling2D, Activation
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.preprocessing.image import load_img
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelBinarizer
import pickle # For saving models to disk and loading saved models
import os
import zipfile
import PIL
import PIL.Image
import pathlib
import cv2
from sklearn.decomposition import PCA
from scipy.stats import stats
import imgaug as ia
ia.seed(1)
from imgaug.augmentables.bbs import BoundingBox, BoundingBoxesOnImage
from imgaug import augmenters as iaa
import imageio
import re
import shutil

load_from_original = True
compress_required = False ### Change to true when running for first time ###

csv_path = os.path.join(capstone_path, "dataframes")
try:
    df_train_ann = pd.read_csv(os.path.join(csv_path, "df_train_ann.csv"))
    df_test_ann = pd.read_csv(os.path.join(csv_path, "df_test_ann.csv"))
    df_train = pd.read_csv(os.path.join(csv_path, "df_train.csv"))
    df_val = pd.read_csv(os.path.join(csv_path, "df_val.csv"))
    df_car_name = pd.read_csv(os.path.join(csv_path, "df_car_name.csv"))
    load_from_original = False
    print("The data will be loaded from saved files!!!")

except:
    print("The data will be loaded from original files!!!")
pass
```

The data will be loaded from saved files!!!

load_from_original

```
False
```

```
### Define various dataframes and constants
BOX_COLOR = (255, 0, 0) # Red
TEXT_COLOR = (0, 0, 0) # White

if load_from_original == True:
    car_img_path = os.path.join(path, 'Car Images/')
    car_ann_path = os.path.join(path, 'Annotations/')
    df_car_name = pd.read_csv("Car names and make.csv",names=['Car Name'])
    train_img_dir = os.path.join(car_img_path, 'Train Images')
    test_img_dir = os.path.join(car_img_path, 'Test Images')
    df_train_ann = pd.read_csv(os.path.join(car_ann_path, 'Train Annotations.csv'),names=['fil
    df_test_ann = pd.read_csv(os.path.join(car_ann_path, 'Test Annotation.csv'),names=['fil
    df_train_ann.drop([df_train_ann.index[0]],inplace=True)
    df_test_ann.drop([df_test_ann.index[0]],inplace=True)
    df_train_ann.reset_index(inplace=True,drop=True)
    df_test_ann.reset_index(inplace=True,drop=True)
    ### Create a dictionary of car names and car classes
    count = 0
    car_dict = {}
    for car_name in df_car_name['Car Name'].values:
        if car_name == 'Ram C/V Cargo Van Minivan 2012':
            car_name = 'Ram C-V Cargo Van Minivan 2012'
        car_dict[count+1] = car_name
        count+=1

    ### Split the car names in to car type, model and year
    car_names = list(df_car_name['Car Name'].values)
    car_types = []
    car_year = []
    car_model = []
    for cars in car_names:
        carname = cars.split(' ')
        model_word_length = len(carname) - 3 #Remove 1 each for name od car company, type
        car_types.append(carname[len(carname)-2])
        car_year.append(carname[len(carname)-1])
        car_model.append(carname[0])
    df_car_name['Car_Type'] = car_types
    df_car_name['Model'] = car_model
    df_car_name['Year'] = car_year
    df_car_name['Car Name'].replace(to_replace=['Ram C/V Cargo Van Minivan 2012'],value='R

df_car_name.sample(10)
```

	Car Name	Car_Type	Model	Year	
19	Audi S6 Sedan 2011	Sedan	Audi	2011	
124	HUMMER H2 SUT Crew Cab 2009	Cab	HUMMER	2009	
40	Bentley Mulsanne Sedan 2011	Sedan	Bentley	2011	
155	Lincoln Town Car Sedan 2011	Sedan	Lincoln	2011	
100	Ferrari FF Coupe 2012	Coupe	Ferrari	2012	
22	BMW M3 Coupe 2012	Coupe	BMW	2012	

```
### Below function to get train/test images path from multiple folders
```

```
### INPUT - parent folder name
```

```
### OUTPUT - a list with full path name for each of the images
```

```
def load_images_from_folder(folder):
```

```
    file_path = {}
```

```
    count = 0
```

```
    for folders in os.listdir(folder):
```

```
        #print(folders)
```

```
        if folders == '.DS_Store':
```

```
            pass
```

```
        else:
```

```
            foldername = os.path.join(folder,folders)
```

```
            #print(foldername)
```

```
            for filename in os.listdir(foldername):
```

```
                count = count+1
```

```
                filepath = os.path.join(foldername,filename)
```

```
                file_path[filename]=filepath
```

```
                #print(filename)
```

```
    return file_path
```

```
### Below function to do PCA transformation of images to reduce dimesionality.
```

```
### INPUT - image and n_components (default is 50)
```

```
### OUTPUT - compressed images returned.
```

```
def compress_pca(image,PCA_n_component=50):
```

```
    img = cv2.imread(image,cv2.COLOR_BGR2RGB)
```

```
    try:
```

```
        if(len(img.shape) < 3):
```

```
            print(len(img.shape))
```

```
            print("\n file_path: ",image)
```

```
            gray = cv2.imread(image,0)
```

```
            blue=red=green=gray
```

```
        else:
```

```
            red,green,blue = cv2.split(img)
```

```
    except:
```

```
        Print("Exception encountered")
```

```
try:
```

```
    df_blue = blue/255
```

```
    df_green = green/255
```

```
    df_red = red/255
```

```
    pca_b = PCA(n_components=PCA_n_component)
```

```
    pca_b.fit(df_blue)
```

```

trans_pca_b = pca_b.transform(df_blue)
pca_g = PCA(n_components=PCA_n_component)
pca_g.fit(df_green)
trans_pca_g = pca_g.transform(df_green)
pca_r = PCA(n_components=PCA_n_component)
pca_r.fit(df_red)
trans_pca_r = pca_r.transform(df_red)
b_arr = pca_b.inverse_transform(trans_pca_b)
g_arr = pca_g.inverse_transform(trans_pca_g)
r_arr = pca_r.inverse_transform(trans_pca_r)
img_reduced= (cv2.merge((r_arr, g_arr, b_arr)))

except:
    return None

return img_reduced

### Below function a wrapper function for compressing a list of images using PCA transform
### INPUT - list of images, path where to store transformed images and n_components for PC
### OUTPUT - output a successful message if completed the operation successfully else retu
def compress_image(image_list,resize_path,PCA_n_component=50):
    filelist = os.listdir(resize_path)
    for img in image_list:
        #print(img)
        img_path = img.split('/')
        if img_path[-1] in filelist:
            pass
        else:
            im = compress_pca(img)
            #print(img_path[-1])
            #if img_path[-1] == "00001.jpg":
            #    plt.imshow(im)
            r_path = os.path.join(resize_path,img_path[-1])
            #print(r_path)
            if im is not None:
                im = cv2.convertScaleAbs(im, alpha=(255.0))
                if not cv2.imwrite(r_path, im):
                    raise Exception("Could not write image")
                cv2.waitKey()
                cv2.destroyAllWindows()
            else:
                print("image skipped:",img_path[-1])
    print("The list of images compressed in folder: ", resize_path)
    compress_required = False
    return

### Below function will calculate the height and width of images path present in a datafra
### INPUT - dataframe and column name of image path column
### OUTPUT - height and width list
def get_image_dims(df,col):
    height = []
    width = []
    for img_path in df[col].values:

```

```



```

```
# Function to convert bounding box image into DataFrame
def bounding_boxes_to_df(bounding_boxes_object):

    # Convert Bounding Boxes Object to Array
    bounding_boxes_array = bounding_boxes_object.to_xyxy_array()

    # Convert the array into DataFrame
    df_bounding_boxes = pd.DataFrame(bounding_boxes_array,
                                      columns=['xmin', 'ymin', 'xmax', 'ymax'])

    # Return the DataFrame
    return df_bounding_boxes

# Define all the Augmentations we want to apply to the dataset
# We're setting random `n` augmentations to 2.
image_augmentations = iaa.SomeOf( 2,
[
    # Scale the Images
    iaa.Affine(scale=(0.5, 1.5)),

    # Rotate the Images
    iaa.Affine(rotate=(-60, 60)),

    # Shift the Image
    iaa.Affine(translate_percent={"x":(-0.3, 0.3), "y":(-0.3, 0.3)}),

    # Flip the Image
    iaa.Fliplr(1),

    # Increase or decrease the brightness
    iaa.Multiply((0.5, 1.5)),

    # Add Gaussian Blur
    iaa.GaussianBlur(sigma=(1.0, 3.0)),

    # Add Gaussian Noise
    iaa.AdditiveGaussianNoise(scale=(0.03*255, 0.05*255))
])

#### Function to augment images and store in the file path input.
def image_aug(df, images_path, aug_images_path, augmentor, multiple = 3):
    # Fill this DataFrame with image attributes
    augmentations_df = pd.DataFrame(columns=['filename', 'width', 'height', 'class', 'xmin',

    # Group the data by filenames
    grouped_df = df.groupby('filename')

    # Create the directory for all augmented images
    if not os.path.exists(aug_images_path):
        os.mkdir(aug_images_path)
```

```

for i in range(multiple):
    # Post Fix we add to the each different augmentation of one image
    image_postfix = str(i)
    # Loop to perform the augmentations
    for filename in df['filename'].unique():
        file_name = filename.split('.')
        fname = file_name[0] + image_postfix + file_name[1]
        augmented_path = os.path.join(aug_images_path, fname)
        # Take one image at a time with its information
        single_image = grouped_df.get_group(filename)
        single_image = single_image.reset_index()
        single_image = single_image.drop(['index'], axis=1)
        # Read the image
        image = imageio.imread(os.path.join(images_path, filename))
        # Get bounding box
        bounding_box_array = single_image.drop(['filename', 'width', 'height',
                                                'class'], axis=1).values
        # Give the bounding box to imgaug library
        bounding_box = BoundingBoxesOnImage.from_xyxy_array(bounding_box_array,
                                                          shape=image.shape)

        # Perform random 2 Augmentations
        image_aug, bounding_box_aug = augmentor(image=image,
                                                bounding_boxes=bounding_box)
        # Discard the the bounding box going out the image completely
        bounding_box_aug = bounding_box_aug.remove_out_of_image()
        # Clip the bounding box that are only partially out of th image
        bounding_box_aug = bounding_box_aug.clip_out_of_image()
        # Get rid of the the image if bounding box was discarded
        if re.findall('Image...', str(bounding_box_aug)) == ['Image([])']:
            pass
        else:
            # Create the augmented image file
            imageio.imwrite(augmented_path, image_aug)
            # Update the image width and height after augmentation
            info_df = single_image.drop(['xmin', 'ymin', 'xmax', 'ymax'], axis=1)
            for index, _ in info_df.iterrows():
                info_df.at[index, 'width'] = image_aug.shape[1]
                info_df.at[index, 'height'] = image_aug.shape[0]
            # Add the prefix to each image to differentiate if required
            info_df['filename'] = info_df['filename'].apply(lambda x: x + image_postfix)
            # Create the augmented bounding boxes dataframe
            bounding_box_df = bounding_boxes_to_df(bounding_box_aug)
            # Concatenate the filenames, height, width and bounding boxes
            aug_df = pd.concat([info_df, bounding_box_df], axis=1)
            # Add all the information to augmentations_df we initialized above
            augmentations_df = pd.concat([augmentations_df, aug_df])

    # Remove index
    augmentations_df = augmentations_df.reset_index()
    augmentations_df = augmentations_df.drop(['index'], axis=1)

    # Return the Dataframe
    return augmentations_df

```

```
#image_directory = os.path.join(os.getcwd(),"resize_size")
```

```
#augmented_images_df = image_aug(df_img_aug, image_directory, 'aug_images_resize',
#                                image_augmentations)

def visualize_bbox(img, bbox, class_name, color=BOX_COLOR, thickness=2):
    """Visualizes a single bounding box on the image"""
    x_min, y_min, x_max, y_max = map(int,bbox)
    cv2.rectangle(img, (x_min, y_min), (x_max, y_max), color=BOX_COLOR, thickness=thickness)
    ((text_width, text_height), _) = cv2.getTextSize(class_name, cv2.FONT_HERSHEY_SIMPLEX, 1, 1)
    cv2.rectangle(img, (x_min, (y_min - int(0.3 * text_height))), (x_min + text_width, y_min))
    cv2.putText(
        img,
        text=class_name,
        org=(x_min, y_min - int(0.3 * text_height)),
        fontFace=cv2.FONT_HERSHEY_SIMPLEX,
        fontScale=0.35,
        color=TEXT_COLOR,
        lineType=cv2.LINE_AA,
    )
    return img

def visualize(image_list, bbox_list, class_list, normalize=False):
    count = 0
    image_final = []
    for image_name in image_list:
        image = cv2.imread(image_name, cv2.COLOR_BGR2RGB)
        img = image.copy()
        h,w,_ = img.shape
        bbox = bbox_list[count]
        if normalize == True:
            bbox[0] = bbox[0]*w
            bbox[1] = bbox[1]*h
            bbox[2] = bbox[2]*w
            bbox[3] = bbox[3]*h
        class_name = class_list[count]
        img = visualize_bbox(img, bbox, str(class_name))
        image_final.append(img)
        count = count + 1
    fig, axs = plt.subplots(int(count), 1, figsize=(70, 70))
    axs = axs.flatten()
    for img,ax in zip(image_final,axs):
        ax.imshow(img)
    plt.show()

# YOLO requires the normalized height and width format.
def YOLO_xywh(xmin, ymin, xmax, ymax, imgw, imgh):
    xmin = float(xmin)
    ymin = float(ymin)
    xmax = float(xmax)
    ymax = float(ymax)
    imgw = float(imgw)
    imgh = float(imgh)
```

```
# print (float(xmin), float(ymin), float(xmax), float(ymax), float(imgw),float(imgh))
xcen = float((xmin + xmax)) / 2 / imgw
ycen = float((ymin + ymax)) / 2 / imgh
w = float((xmax - xmin)) / imgw
h = float((ymax - ymin)) / imgh
return xcen, ycen, w, h
```

df_train_ann

	filename	width	height	class	xmin	xmax	ymin	ymax	class_name
0	00001.jpg	600	400	14	39.0	569.0	116.0	375.0	Audi TTS Coupe 2012
1	00002.jpg	900	675	3	36.0	868.0	116.0	587.0	Acura TL Sedan 2012
2	00003.jpg	640	480	91	85.0	601.0	109.0	381.0	Dodge Dakota Club Cab 2007
3	00004.jpg	2100	1386	134	621.0	1484.0	393.0	1096.0	Hyundai Sonata Hybrid Sedan 2012
4	00005.jpg	144	108	106	14.0	133.0	36.0	99.0	Ford F-450 Super Duty Crew Cab 2012

```
if load_from_original == True:
    train_images_path = load_images_from_folder(train_img_dir)
    test_images_path = load_images_from_folder(test_img_dir)
    df_train_ann['filepath'] = df_train_ann['filename'].apply(lambda x:train_images_path[x])
    df_train_ann['class_name'] = df_train_ann['class'].apply(lambda x:car_dict[int(x)])
    df_train_ann['xmin'] = df_train_ann.xmin.astype(float)
    #df_train_ann['class'] = df_train_ann['class'].astype(int)
    #ds_train_ann['Bbox'] = ds_train_ann.apply(lambda x:list([x['Start_x'],x['Start_y'],x['End_x'],x['End_y']]))
    df_test_ann['filepath'] = df_test_ann['filename'].apply(lambda x:test_images_path[x])
    df_test_ann['xmin'] = df_test_ann.xmin.astype(float)
    #df_test_ann['class'] = df_test_ann['class'].astype(int)
    df_test_ann['class_name'] = df_test_ann['class'].apply(lambda x:car_dict[int(x)])
    #df_test_ann['Bbox'] = ds_test_ann.apply(lambda x:list([x['Start_x'],x['Start_y'],x['End_x'],x['End_y']]))
```

```
if load_from_original == True:
    trainh,trainw = get_image_dims(df_train_ann,'filepath')
    df_train_ann['width'] = trainw
    df_train_ann['height'] = trainh
    testh,testw = get_image_dims(df_test_ann,'filepath')
    df_test_ann['width'] = testw
```

```
df_test_ann['height'] = testh
col_list = ['filename','width','height','class','xmin','xmax','ymin','ymax','class_name']
df_train_ann = df_train_ann[col_list]
df_test_ann = df_test_ann[col_list]
```

```
df_train_ann.sample(5)
```

		filename	width	height	class	xmin	xmax	ymin	ymax	class_name
3249	03250.jpg		640	426	176	366.0	602.0	163.0	301.0	Rolls-Royce Ghost Sedan 2012
1072	01073.jpg		555	370	24	8.0	545.0	63.0	314.0	Audi S4 Sedan 2007
2060	02061.jpg		688	456	183	43.0	646.0	110.0	427.0	Suzuki SX4 Hatchback

```
df_test_ann.sample(5)
```

		filename	width	height	class	xmin	xmax	ymin	ymax	class_name
2566	02567.jpg		495	223	152	42.0	471.0	65.0	197.0	Lamborghini Gallardo LP 570-4 Superleggera 2012
7898	07899.jpg		300	225	148	28.0	282.0	18.0	183.0	Jeep Grand Cherokee SUV 2012

```
from sklearn.model_selection import train_test_split
if load_from_original == True:
    df_train,df_val = train_test_split(df_train_ann,test_size=0.2,random_state=1,stratify=df_train['class_name'])
```

```
df_train.shape
```

```
(6515, 10)
```

```
df_train.sample(5)
```

		filename	width	height	class	xmin	xmax	ymin	ymax	class_name
6002	07010.jpg		560	373	120	118.0	485.0	112.0	351.0	GMC Yukon Hybrid SUV 2012
2154	05689.jpg		655	261	132	27.0	509.0	8.0	255.0	Hyundai Tucson SUV 2012
6194	01758.jpg		640	480	114	1.0	639.0	103.0	480.0	Ford F-150 Regular

```
df_val.shape
```

(1629, 10)

```
df_val.sample(5)
```

	filename	width	height	class	xmin	xmax	ymin	ymax	class_name
717	00478.jpg	112	84	174	9.0	98.0	7.0	81.0	Ram C-V Cargo Van Minivan 2012
238	01456.jpg	500	375	120	35.0	464.0	63.0	331.0	GMC Yukon Hybrid SUV 2012

```
print(list(df_train['class'].value_counts()))
print(list(df_val['class'].value_counts()))
print(list(df_test_ann['class'].value_counts()))
```

```
print(len(list(df_train['class'].value_counts())))
print(len(list(df_val['class'].value_counts())))
print(len(list(df_test_ann['class'].value_counts())))
```

196

if load from original == True:

```
csv_path = os.path.join(capstone_path,"dataframes")
df_train_ann.to_csv(os.path.join(csv_path,"df_train_ann.csv"),index=None)
df_test_ann.to_csv(os.path.join(csv_path,"df_test_ann.csv"),index=None)
df_train.to_csv(os.path.join(csv_path,"df_train.csv"),index=None)
df_val.to_csv(os.path.join(csv_path,"df_val.csv"),index=None)
df_car_name.to_csv(os.path.join(csv_path,"df_car_name.csv"),index=None)
load_from_original = False
```

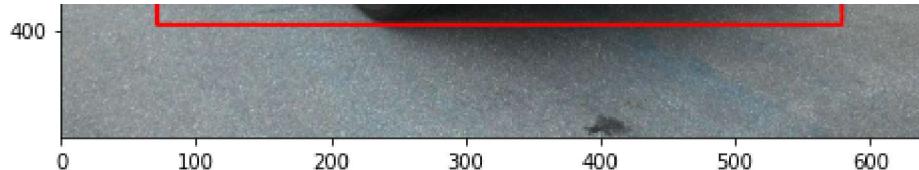
```
df_visualize = df_train_ann[['filepath','class_name','xmin','ymin','xmax','ymax']].sample(1)
df_visualize['Bbox'] = df_visualize.apply(lambda x:list([float(x['xmin']),float(x['ymin'])]),axis=1)
df_visualize.drop(columns=['xmin','ymin','xmax','ymax'],inplace=True)
```

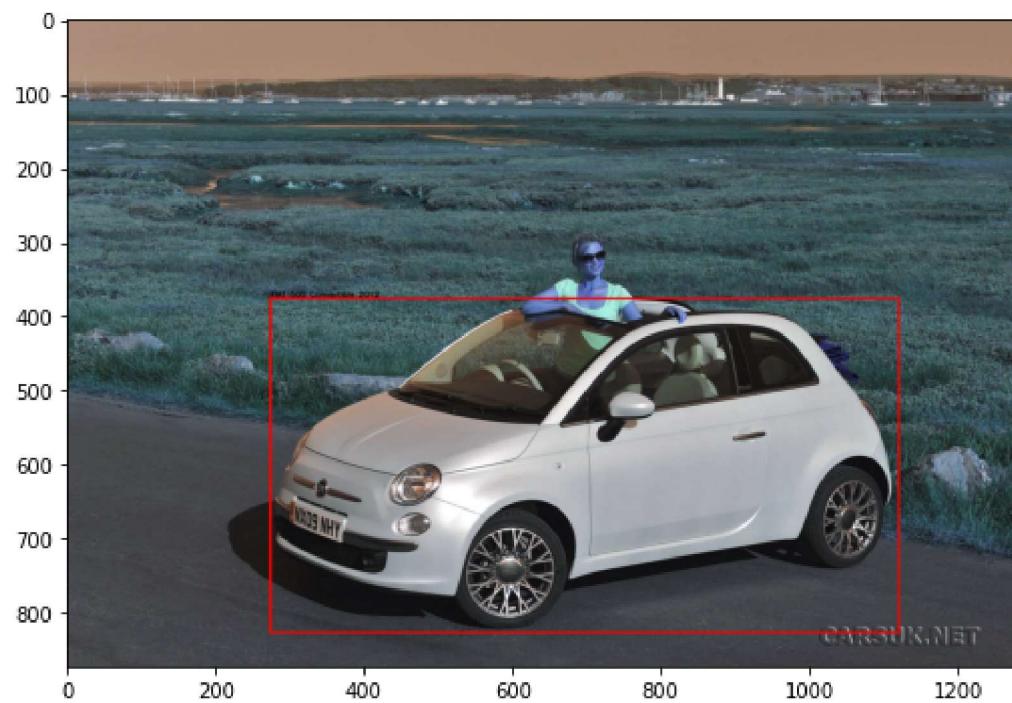
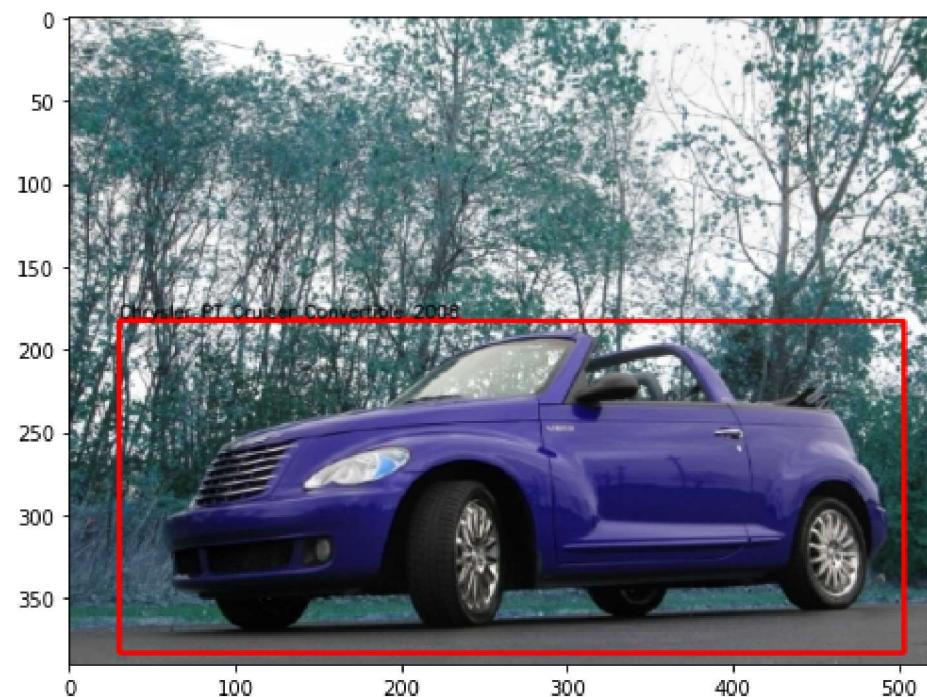
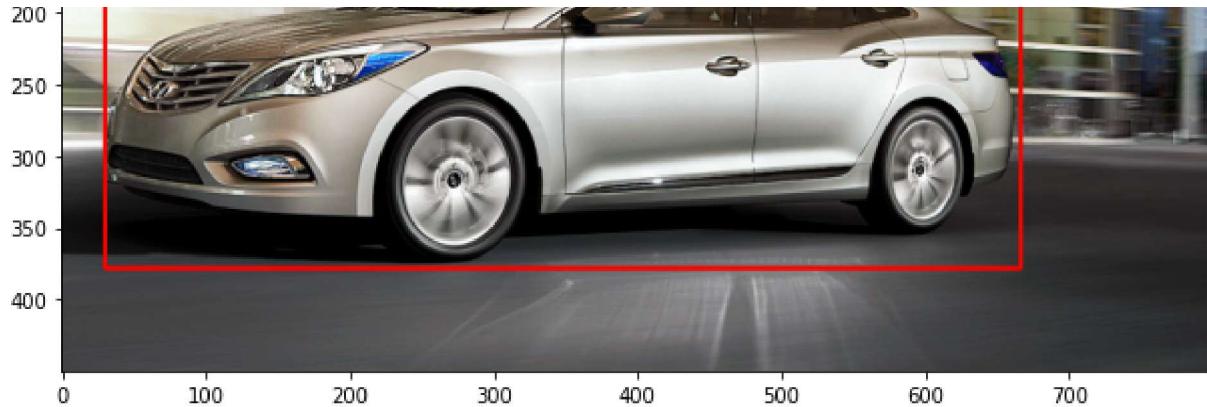
df visualize

	filepath	class_name	Bbox
4299	/content/drive/My Drive/capstone/Dataset/Car I...	Chrysler Aspen SUV 2009	[16.0, 29.0, 241.0, 155.0]
2723	/content/drive/My Drive/capstone/Dataset/Car I...	Ford Freestar Minivan 2007	[38.0, 18.0, 192.0, 110.0]
4127	/content/drive/My Drive/capstone/Dataset/Car I...	Volkswagen Beetle Hatchback 2012	[71.0, 65.0, 579.0, 396.0]
1381	/content/drive/My Drive/capstone/Dataset/Car I...	Cadillac Escalade EXT Crew Cab 2007	[3.0, 21.0, 285.0, 195.0]
1800	/content/drive/My Drive/capstone/Dataset/Car I...	Dodge Durango SUV 2007	[15.0, 25.0, 239.0, 182.0]
8005	/content/drive/My Drive/capstone/Dataset/Car I...	Hyundai Azera Sedan 2012	[30.0, 146.0, 666.0, 378.01]

```
image_list = list(df_visualize['filepath'])
class_list = list(df_visualize['class_name'])
bbox_list = list(df_visualize['Bbox'])
visualize(image_list,bbox_list,class_list)
```









```
if compress_required == True:  
    comp_image_path = os.path.join(capstone_path, "compressed_images")  
    img_list = list(df_train["filepath"])  
    compress_image(img_list, comp_image_path)  
    compress_required = True  
    comp_image_path = os.path.join(capstone_path, "compressed_val_images")  
    img_list = list(df_val["filepath"])  
    compress_image(img_list, comp_image_path)  
    compress_required = True  
    comp_image_path = os.path.join(capstone_path, "compressed_test_images")  
    img_list = list(df_test_ann["filepath"])  
    compress_image(img_list, comp_image_path)  
    compress_required = False
```

```
!python -c "import tensorflow as tf;print(tf.reduce_sum(tf.random.normal([1000, 1000])))"
```

```
2022-01-30 08:03:18.822503: W tensorflow/core/common_runtime/gpu/gpu_bfc_allocator.cc:101] Allocator (0x7f3e00000000) failed to allocate 1019.2014 bytes with 1024 at host with 1024 available free bytes.  
tf.Tensor(-1019.2014, shape=(), dtype=float32)
```

```
os.chdir(capstone_path)
```

```
%cd tensorflow
```

```
/content/drive/My Drive/capstone/tensorflow
```

```
#!git clone https://github.com/tensorflow/models.git
```

```
%cd ./models/research/
```

```
/content/drive/My Drive/capstone/tensorflow/models/research
```

```
!protoc object_detection/protos/*.proto --python_out=.
```

```
#!git clone https://github.com/cocodataset/cocoapi.git
```

```
#%cd cocoapi/PythonAPI
```

```
#!make
```

```
#!cp -r pycocotools ../../
```

```
!pwd
```

```
/content/drive/MyDrive/capstone/tensorflow/models/research
```

```
#cd workspace/training-yolo/
```

```
#cd yolov5
```

```
!ls
```

adversarial_text	deep_speech	pcl_rl
attention_ocr	delf	pycocotools
audioset	efficient-hrl	README.md
autoaugment	lfads	rebar
build	lstm_object_detection	seq_flow_lite
cocoapi	marco	setup.py
cognitive_planning	nst_blogpost	slim
cvt_text	object_detection	vid2depth
deeplab	object_detection.egg-info	

```
!pwd
```

```
/content/drive/MyDrive/capstone/tensorflow/models/research
```

```
!cp object_detection/packages/tf2/setup.py .
```

```
#!python setup.py install
```

```
!python -m pip install --use-feature=in-tree-build .
```

```
Requirement already satisfied: wheel<1.0,>=0.32.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: keras-preprocessing>=1.1.1 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: tensorboard~>2.6 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: cached-property in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: dm-tree~>0.1.1 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: fastavro<2,>=0.21.4 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: pymongo<4.0.0,>=3.8.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: pyarrow<7.0.0,>=0.15.1 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: proto-plus<2,>=1.7.1 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: hdfs<3.0.0,>=2.1.0 in /usr/local/lib/python3.7/dist-packages
```

```
Requirement already satisfied: crcmod<2.0,>=1.7 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: pydot<2,>=1.2.0 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: dill<0.3.2,>=0.3.1.1 in /usr/local/lib/python3.7/di
Requirement already satisfied: orjson<4.0 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: docopt in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: opencv-python>=4.1.0.25 in /usr/local/lib/python3.7
Requirement already satisfied: kiwisolver>=1.1.0 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: cycler>=0.10.0 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.7/dis
Requirement already satisfied: tabulate>=0.8.9 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: regex in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: colorama in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: portalocker in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: scikit-learn>=0.21.3 in /usr/local/lib/python3.7/di
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/di
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: typeguard>=2.7 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: attrs>=18.1.0 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: tensorflow-metadata in /usr/local/lib/python3.7/dis
Requirement already satisfied: promise in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: importlib-resources in /usr/local/lib/python3.7/dis
Requirement already satisfied: future in /usr/local/lib/python3.7/dist-packages (f
Building wheels for collected packages: object-detection
  Building wheel for object-detection (setup.py) ... done
    Created wheel for object-detection: filename=object_detection-0.1-py3-none-any.w
      Stored in directory: /tmp/pip-ephem-wheel-cache-jucvrkqa/wheels/3a/7d/30/e990abb
Successfully built object-detection
Installing collected packages: object-detection
  Attempting uninstall: object-detection
    Found existing installation: object-detection 0.1
      Uninstalling object-detection-0.1:
        Successfully uninstalled object-detection-0.1
Successfully installed object-detection-0.1
```

```
!python object_detection/builders/model_builder_tf2_test.py
```

```
I0130 08:06:14.322106 140098475657088 ssd_efficientnet_bifpn_feature_extractor.py: 
I0130 08:06:14.322283 140098475657088 ssd_efficientnet_bifpn_feature_extractor.py: 
I0130 08:06:14.322356 140098475657088 ssd_efficientnet_bifpn_feature_extractor.py: 
I0130 08:06:14.323943 140098475657088 efficientnet_model.py:147] round_filter inpu
I0130 08:06:14.338570 140098475657088 efficientnet_model.py:147] round_filter inpu
I0130 08:06:14.338687 140098475657088 efficientnet_model.py:147] round_filter inpu
I0130 08:06:14.561251 140098475657088 efficientnet_model.py:147] round_filter inpu
I0130 08:06:14.561438 140098475657088 efficientnet_model.py:147] round_filter inpu
I0130 08:06:15.048639 140098475657088 efficientnet_model.py:147] round_filter inpu
I0130 08:06:15.048818 140098475657088 efficientnet_model.py:147] round_filter inpu
I0130 08:06:15.528684 140098475657088 efficientnet_model.py:147] round_filter inpu
I0130 08:06:15.528866 140098475657088 efficientnet_model.py:147] round_filter inpu
I0130 08:06:16.230118 140098475657088 efficientnet_model.py:147] round_filter inpu
I0130 08:06:16.230304 140098475657088 efficientnet_model.py:147] round_filter inpu
I0130 08:06:16.929373 140098475657088 efficientnet_model.py:147] round_filter inpu
I0130 08:06:16.929553 140098475657088 efficientnet_model.py:147] round_filter inpu
I0130 08:06:18.075128 140098475657088 efficientnet_model.py:147] round_filter inpu
I0130 08:06:18.075303 140098475657088 efficientnet_model.py:147] round_filter inpu
I0130 08:06:18.362827 140098475657088 efficientnet_model.py:147] round_filter inpu
I0130 08:06:18.389021 140098475657088 efficientnet_model.py:457] Building model ef
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_create_ssd_models_from_conf
I0130 08:06:18.504767 140098475657088 test_util.py:2309] time(__main__.ModelBuilde
[ OK ] ModelBuilderTF2Test.test create ssd models from config
```

```
[ RUN      ] ModelBuilderTF2Test.test_invalid_faster_rcnn_batchnorm_update
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_invalid_faster_rcnn_batchnc
I0130 08:06:18.512332 140098475657088 test_util.py:2309] time(__main__.ModelBuilde
[       OK ] ModelBuilderTF2Test.test_invalid_faster_rcnn_batchnorm_update
[ RUN      ] ModelBuilderTF2Test.test_invalid_first_stage_nms_iou_threshold
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_invalid_first_stage_nms_iou
I0130 08:06:18.514171 140098475657088 test_util.py:2309] time(__main__.ModelBuilde
[       OK ] ModelBuilderTF2Test.test_invalid_first_stage_nms_iou_threshold
[ RUN      ] ModelBuilderTF2Test.test_invalid_model_config_proto
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_invalid_model_config_proto)
I0130 08:06:18.514730 140098475657088 test_util.py:2309] time(__main__.ModelBuilde
[       OK ] ModelBuilderTF2Test.test_invalid_model_config_proto
[ RUN      ] ModelBuilderTF2Test.test_invalid_second_stage_batch_size
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_invalid_second_stage_batch_
I0130 08:06:18.516406 140098475657088 test_util.py:2309] time(__main__.ModelBuilde
[       OK ] ModelBuilderTF2Test.test_invalid_second_stage_batch_size
[ RUN      ] ModelBuilderTF2Test.test_session
[  SKIPPED ] ModelBuilderTF2Test.test_session
[ RUN      ] ModelBuilderTF2Test.test_unknown_faster_rcnn_feature_extractor
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_unknown_faster_rcnn_feature
I0130 08:06:18.517955 140098475657088 test_util.py:2309] time(__main__.ModelBuilde
[       OK ] ModelBuilderTF2Test.test_unknown_faster_rcnn_feature_extractor
[ RUN      ] ModelBuilderTF2Test.test_unknown_meta_architecture
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_unknown_meta_architecture):
I0130 08:06:18.518465 140098475657088 test_util.py:2309] time(__main__.ModelBuilde
[       OK ] ModelBuilderTF2Test.test_unknown_meta_architecture
[ RUN      ] ModelBuilderTF2Test.test_unknown_ssd_feature_extractor
INFO:tensorflow:time(__main__.ModelBuilderTF2Test.test_unknown_ssd_feature_extract
I0130 08:06:18.519565 140098475657088 test_util.py:2309] time(__main__.ModelBuilde
[       OK ] ModelBuilderTF2Test.test_unknown_ssd_feature_extractor
-----
```

Ran 24 tests in 24.399s

OK (skipped=1)

▼ SSD Mobilenet based object detection algorithm

```
#%cd ..
#%cd ..
```

```
#%cd workspace/training-ssd
```

```
ssd_path = os.path.join(capstone_path, "tensorflow/workspace/training-ssd")
os.chdir(ssd_path)
```

```
df_train.sample(5)
```

filename	width	height	class	xmin	xmax	ymin	ymax	class_name	
3803	04961.jpg	750	469	154	84.0	686.0	89.0	424.0	Land Rover Range Rover SUV 2012

```

train_csv = df_train[['filename','width','height','class_name','xmin','ymin','xmax','ymax']]
val_csv = df_val[['filename','width','height','class_name','xmin','ymin','xmax','ymax']]

path_to_pbtext = os.path.join(ssd_path,"annotations/label_map.pbtxt")
path_tfrecord_train = os.path.join(ssd_path,"annotations/train.record")
path_tfrecord_test = os.path.join(ssd_path,"annotations/test.record")
path_to_train_images = os.path.join(ssd_path,"images/compressed_images")
path_to_test_images = os.path.join(ssd_path,"images/compressed_test_images")
path_to_train_csv = os.path.join(capstone_path,"train_csv.csv")
path_to_test_csv = os.path.join(capstone_path,"val_csv.csv")

#train_csv.to_csv(path_to_train_csv,index=None)
#val_csv.to_csv(path_to_test_csv,index=None)

train_csv = pd.read_csv(path_to_train_csv)
val_csv = pd.read_csv(path_to_test_csv)

%cd ..
/content/drive/MyDrive/capstone/tensorflow/workspace

%cd ..
/content/drive/MyDrive/capstone/tensorflow

%cd /content/drive/My Drive/capstone/tensorflow/scripts
/content/drive/My Drive/capstone/tensorflow/scripts

#!python generate_tfrecord.py -x ../workspace/training-ssd/images/compressed_images -l ../

#!python generate_tfrecord.py -x ../workspace/training-ssd/images/compressed_val_images -l

!ls
Annotations 'Car Images' 'Car names and make.csv' __MACOSX

%cd /content/drive/My Drive/capstone/tensorflow/workspace/training-ssd/
/content/drive/My Drive/capstone/tensorflow/workspace/training-ssd

```

```
!ls
```

```
annotations           images          'pre-trained models'  
exported-models      model_main_tf2.py  
exporter_main_v2.py   models
```

```
!python model_main_tf2.py --model_dir=models/my_ssd_mobilenet --pipeline_config_path=model
```

```
2022-01-23 12:20:13.368157: W tensorflow/core/common_runtime/gpu/gpu_bfc_allocator  
INFO:tensorflow:Using MirroredStrategy with devices ('/job:localhost/replica:0/tas  
I0123 12:20:13.371561 139706229311360 mirrored_strategy.py:376] Using MirroredStra  
INFO:tensorflow:Maybe overwriting train_steps: None  
I0123 12:20:14.230054 139706229311360 config_util.py:552] Maybe overwriting train_  
INFO:tensorflow:Maybe overwriting use_bfloat16: False  
I0123 12:20:14.230258 139706229311360 config_util.py:552] Maybe overwriting use_bf  
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/object_detection/mc  
Instructions for updating:  
rename to distribute_datasets_from_function  
W0123 12:20:14.259042 139706229311360 deprecation.py:347] From /usr/local/lib/pyth  
Instructions for updating:  
rename to distribute_datasets_from_function  
INFO:tensorflow:Reading unweighted datasets: ['./annotations/train.record']  
I0123 12:20:14.674426 139706229311360 dataset_builder.py:163] Reading unweighted d  
INFO:tensorflow:Reading record datasets for input file: ['./annotations/train.recc  
I0123 12:20:14.674803 139706229311360 dataset_builder.py:80] Reading record database  
INFO:tensorflow:Number of filenames to read: 1  
I0123 12:20:14.674925 139706229311360 dataset_builder.py:81] Number of filenames t  
WARNING:tensorflow:num_readers has been reduced to 1 to match input file shards.  
W0123 12:20:14.675002 139706229311360 dataset_builder.py:88] num_readers has been  
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/object_detection/bu  
Instructions for updating:  
Use `tf.data.Dataset.interleave(map_func, cycle_length, block_length, num_parallel  
W0123 12:20:14.681763 139706229311360 deprecation.py:347] From /usr/local/lib/pyth  
Instructions for updating:  
Use `tf.data.Dataset.interleave(map_func, cycle_length, block_length, num_parallel  
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/object_detection/bu  
Instructions for updating:  
Use `tf.data.Dataset.map()  
W0123 12:20:14.709128 139706229311360 deprecation.py:347] From /usr/local/lib/pyth  
Instructions for updating:  
Use `tf.data.Dataset.map()  
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/python/u  
Instructions for updating:  
Create a `tf.sparse.SparseTensor` and use `tf.sparse.to_dense` instead.  
W0123 12:20:21.463398 139706229311360 deprecation.py:347] From /usr/local/lib/pyth  
Instructions for updating:  
Create a `tf.sparse.SparseTensor` and use `tf.sparse.to_dense` instead.  
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/python/u  
Instructions for updating:  
`seed2` arg is deprecated. Use sample_distorted_bounding_box_v2 instead.  
W0123 12:20:24.445254 139706229311360 deprecation.py:347] From /usr/local/lib/pyth  
Instructions for updating:  
`seed2` arg is deprecated. Use sample_distorted_bounding_box_v2 instead.  
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/python/a  
Instructions for updating:  
Use `tf.cast` instead.  
W0123 12:20:26.018429 139706229311360 deprecation.py:347] From /usr/local/lib/pyth  
Instructions for updating:  
Use `tf.cast` instead.
```

```
2022-01-23 12:20:32.411990: W tensorflow/core/framework/op_kernel.cc:1745] OP_REQUESTED /usr/local/lib/python3.7/dist-packages/keras/backend.py:414: UserWarning: `tf.keras.warnings.warn('`tf.keras.backend.set_learning_phase` is deprecated and' WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/python/u Instructions for updating: Use fn_output_signature instead
```

```
!python model_main_tf2.py --model_dir=models/my_ssd_mobilenet --pipeline_config_path=model
```

```
Traceback (most recent call last):
  File "model_main_tf2.py", line 32, in <module>
    from object_detection import model_lib_v2
ModuleNotFoundError: No module named 'object_detection'
```

```
!pwd
```

```
/content/drive/MyDrive/capstone/tensorflow/workspace/training-ssd
```

```
!python ./exporter_main_v2.py --input_type image_tensor --pipeline_config_path ./models/my
```

```
W0123 13:48:40.182022 139807483082624 save_impl.py:[2] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.core.L
W0123 13:48:40.182077 139807483082624 save_impl.py:[2] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.cc>
W0123 13:48:40.182204 139807483082624 save_impl.py:[2] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.core.L
W0123 13:48:40.182262 139807483082624 save_impl.py:[2] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.cc>
W0123 13:48:40.182318 139807483082624 save_impl.py:[2] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.core.L
W0123 13:48:40.182373 139807483082624 save_impl.py:[2] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.cc>
W0123 13:48:40.182427 139807483082624 save_impl.py:[2] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.core.L
W0123 13:48:40.182485 139807483082624 save_impl.py:[2] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.cc>
W0123 13:48:40.182540 139807483082624 save_impl.py:[2] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.core.L
W0123 13:48:40.182594 139807483082624 save_impl.py:[2] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.cc>
W0123 13:48:40.182648 139807483082624 save_impl.py:[2] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.core.L
W0123 13:48:40.182706 139807483082624 save_impl.py:[2] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.cc>
W0123 13:48:40.182767 139807483082624 save_impl.py:[2] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.core.L
W0123 13:48:40.182822 139807483082624 save_impl.py:[2] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.cc>
W0123 13:48:40.182875 139807483082624 save_impl.py:[2] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.core.L
W0123 13:48:40.182928 139807483082624 save_impl.py:[2] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.cc>
W0123 13:48:40.182985 139807483082624 save_impl.py:[2] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.core.L
W0123 13:48:40.183042 139807483082624 save_impl.py:[2] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.cc>
W0123 13:48:40.183103 139807483082624 save_impl.py:[2] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.core.L
```

```
W0123 13:48:40.183159 139807483082624 save_impl.py:72] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.cc>
W0123 13:48:40.183212 139807483082624 save_impl.py:72] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.core.Layer>
W0123 13:48:40.183266 139807483082624 save_impl.py:72] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.cc>
W0123 13:48:40.183320 139807483082624 save_impl.py:72] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.core.Layer>
W0123 13:48:40.183383 139807483082624 save_impl.py:72] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.cc>
W0123 13:48:40.183437 139807483082624 save_impl.py:72] Skipping full serialization
WARNING:tensorflow:Skipping full serialization of Keras layer <keras.layers.core.Layer>
W0123 13:48:40.232452 139807483082624 save_impl.py:72] Skipping full serialization
2022-01-23 13:48:49.249368: W tensorflow/python/util/util.cc:368] Sets are not cur
W0123 13:49:27.565280 139807483082624 save.py:268] Found untraced functions such a
INFO:tensorflow:Assets written to: ./exported-models/my_model/saved_model/assets
I0123 13:49:36.735169 139807483082624 builder_impl.py:784] Assets written to: ./ex
INFO:tensorflow:Writing pipeline config file to ./exported-models/my_model/pipelin
I0123 13:49:37.740982 139807483082624 config_util.py:254] Writing pipeline config
```

▼ Faster RCNN Object Detection model

```
!python model_main_tf2.py --model_dir=models/my_rcnn_imagenet --pipeline_config_path=model_main.config
{'learning_rate': 0.014529499}
I0123 16:00:20.294896 140287364614016 model_lib_v2.py:708] {'Loss/BoxClassifierLoss': 0.0,
'Loss/BoxClassifierLoss/localization_loss': 0.0,
'Loss/RPNLoss/localization_loss': 57.503975,
'Loss/RPNLoss/objectness_loss': 0.20698807,
'Loss/regularization_loss': 0.0,
'Loss/total_loss': 57.711205,
'learning_rate': 0.014529499}
INFO:tensorflow:Step 81100 per-step time 0.418s
I0123 16:01:02.056276 140287364614016 model_lib_v2.py:707] Step 81100 per-step tim
INFO:tensorflow:{'Loss/BoxClassifierLoss/classification_loss': 0.11626255,
'Loss/BoxClassifierLoss/localization_loss': 0.0,
'Loss/RPNLoss/localization_loss': 6.1249623,
'Loss/RPNLoss/objectness_loss': 2.226023,
'Loss/regularization_loss': 0.0,
'Loss/total_loss': 8.467248,
'learning_rate': 0.014381699}
I0123 16:01:02.056622 140287364614016 model_lib_v2.py:708] {'Loss/BoxClassifierLoss': 0.0,
'Loss/BoxClassifierLoss/localization_loss': 0.0,
'Loss/RPNLoss/localization_loss': 6.1249623,
'Loss/RPNLoss/objectness_loss': 2.226023,
'Loss/regularization_loss': 0.0,
'Loss/total_loss': 8.467248,
'learning_rate': 0.014381699}
INFO:tensorflow:Step 81200 per-step time 0.385s
I0123 16:01:40.548032 140287364614016 model_lib_v2.py:707] Step 81200 per-step tim
INFO:tensorflow:{'Loss/BoxClassifierLoss/classification_loss': 0.008738259,
'Loss/BoxClassifierLoss/localization_loss': 0.0,
'Loss/RPNLoss/localization_loss': 36.51449,
'Loss/RPNLoss/objectness_loss': 0.5205713,
'Loss/regularization_loss': 0.0,
'Loss/total loss': 37.043797,
```

```
'learning_rate': 0.014234595}
I0123 16:01:40.548349 140287364614016 model_lib_v2.py:708] {'Loss/BoxClassifierLoss/loss': 0.0,
'Loss/BoxClassifierLoss/localization_loss': 0.0,
'Loss/RPNLoss/localization_loss': 36.51449,
'Loss/RPNLoss/objectness_loss': 0.5205713,
'Loss/regularization_loss': 0.0,
'Loss/total_loss': 37.043797,
'learning_rate': 0.014234595}

INFO:tensorflow:Step 81300 per-step time 0.385s
I0123 16:02:19.061013 140287364614016 model_lib_v2.py:707] Step 81300 per-step tim
INFO:tensorflow:{'Loss/BoxClassifierLoss/classification_loss': 0.00025892313,
'Loss/BoxClassifierLoss/localization_loss': 0.0,
'Loss/RPNLoss/localization_loss': 0.42648917,
'Loss/RPNLoss/objectness_loss': 0.54572624,
'Loss/regularization_loss': 0.0,
'Loss/total_loss': 0.97247434,
'learning_rate': 0.014088149}

I0123 16:02:19.061323 140287364614016 model_lib_v2.py:708] {'Loss/BoxClassifierLos
'Loss/BoxClassifierLoss/localization_loss': 0.0,
'Loss/RPNLoss/localization_loss': 0.42648917,
'Loss/RPNLoss/objectness_loss': 0.54572624,
'Loss/regularization_loss': 0.0,
'Loss/total_loss': 0.97247434,
'learning_rate': 0.014088149}

^C
```

```
!python model_main_tf2.py --model_dir=models/my_rcnn_imagenet --pipeline_config_path=model
running per image evaluation...
Evaluate annotation type *bbox*
DONE (t=3.93s).
Accumulating evaluation results...
DONE (t=0.91s).
Average Precision (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.000
Average Precision (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.000
Average Precision (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.000
Average Precision (AP) @[ IoU=0.50:0.95 | area= small  | maxDets=100 ] = -1.000
Average Precision (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.000
Average Precision (AP) @[ IoU=0.50:0.95 | area= large  | maxDets=100 ] = 0.000
Average Recall    (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 1 ] = 0.000
Average Recall    (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.000
Average Recall    (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.000
Average Recall    (AR) @[ IoU=0.50:0.95 | area= small  | maxDets=100 ] = -1.000
Average Recall    (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.000
Average Recall    (AR) @[ IoU=0.50:0.95 | area= large  | maxDets=100 ] = 0.000
INFO:tensorflow:Eval metrics at step 81000
I0124 14:50:46.247649 140299072845696 model_lib_v2.py:1015] Eval metrics at step 8
INFO:tensorflow:      + DetectionBoxes_Precision/mAP: 0.000000
I0124 14:50:46.262341 140299072845696 model_lib_v2.py:1018]      + DetectionBoxes_F
INFO:tensorflow:      + DetectionBoxes_Precision/mAP@.50IOU: 0.000000
I0124 14:50:46.263627 140299072845696 model_lib_v2.py:1018]      + DetectionBoxes_F
INFO:tensorflow:      + DetectionBoxes_Precision/mAP@.75IOU: 0.000000
I0124 14:50:46.264754 140299072845696 model_lib_v2.py:1018]      + DetectionBoxes_F
INFO:tensorflow:      + DetectionBoxes_Precision/mAP (small): -1.000000
I0124 14:50:46.265940 140299072845696 model_lib_v2.py:1018]      + DetectionBoxes_F
INFO:tensorflow:      + DetectionBoxes_Precision/mAP (medium): 0.000000
I0124 14:50:46.266924 140299072845696 model_lib_v2.py:1018]      + DetectionBoxes_F
INFO:tensorflow:      + DetectionBoxes_Precision/mAP (large): 0.000000
I0124 14:50:46.267962 140299072845696 model lib v2.py:1018]      + DetectionBoxes F
```

```

INFO:tensorflow:    + DetectionBoxes_Recall/AR@1: 0.000000
I0124 14:50:46.269064 140299072845696 model_lib_v2.py:1018]    + DetectionBoxes_F
INFO:tensorflow:    + DetectionBoxes_Recall/AR@10: 0.000000
I0124 14:50:46.270041 140299072845696 model_lib_v2.py:1018]    + DetectionBoxes_F
INFO:tensorflow:    + DetectionBoxes_Recall/AR@100: 0.000000
I0124 14:50:46.271005 140299072845696 model_lib_v2.py:1018]    + DetectionBoxes_F
INFO:tensorflow:    + DetectionBoxes_Recall/AR@100 (small): -1.000000
I0124 14:50:46.271989 140299072845696 model_lib_v2.py:1018]    + DetectionBoxes_F
INFO:tensorflow:    + DetectionBoxes_Recall/AR@100 (medium): 0.000000
I0124 14:50:46.273025 140299072845696 model_lib_v2.py:1018]    + DetectionBoxes_F
INFO:tensorflow:    + DetectionBoxes_Recall/AR@100 (large): 0.000000
I0124 14:50:46.274218 140299072845696 model_lib_v2.py:1018]    + DetectionBoxes_F
INFO:tensorflow:    + Loss/RPNLoss/localization_loss: 39.191086
I0124 14:50:46.275136 140299072845696 model_lib_v2.py:1018]    + Loss/RPNLoss/loc
INFO:tensorflow:    + Loss/RPNLoss/objectness_loss: 0.310603
I0124 14:50:46.276059 140299072845696 model_lib_v2.py:1018]    + Loss/RPNLoss/obj
INFO:tensorflow:    + Loss/BoxClassifierLoss/localization_loss: 0.000000
I0124 14:50:46.276993 140299072845696 model_lib_v2.py:1018]    + Loss/BoxClassifi
INFO:tensorflow:    + Loss/BoxClassifierLoss/classification_loss: 0.000000
I0124 14:50:46.277999 140299072845696 model_lib_v2.py:1018]    + Loss/BoxClassifi
INFO:tensorflow:    + Loss/regularization_loss: 0.000000
I0124 14:50:46.279058 140299072845696 model_lib_v2.py:1018]    + Loss/regularizat
INFO:tensorflow:    + Loss/total_loss: 39.501694
I0124 14:50:46.280007 140299072845696 model_lib_v2.py:1018]    + Loss/total_loss:
INFO:tensorflow:Waiting for new checkpoint at models/my_rcnn_imagenet
I0124 14:50:47.107184 140299072845696 checkpoint_utils.py:140] Waiting for new che

```

!python ./exporter_main_v2.py --input_type image_tensor --pipeline_config_path ./models/my

```

2022-01-24 15:34:34.650881: W tensorflow/core/common_runtime/gpu/gpu_bfc_allocator.cc
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/python/autc
Instructions for updating:
back_prop=False is deprecated. Consider using tf.stop_gradient instead.
Instead of:
results = tf.map_fn(fn, elems, back_prop=False)
Use:
results = tf.nest.map_structure(tf.stop_gradient, tf.map_fn(fn, elems))
W0124 15:34:36.505764 140620629907328 deprecation.py:619] From /usr/local/lib/python:
Instructions for updating:
back_prop=False is deprecated. Consider using tf.stop_gradient instead.
Instead of:
results = tf.map_fn(fn, elems, back_prop=False)
Use:
results = tf.nest.map_structure(tf.stop_gradient, tf.map_fn(fn, elems))
INFO:tensorflow:depth of additional conv before box predictor: 0
I0124 15:34:52.334997 140620629907328 convolutional_keras_box_predictor.py:154] dept
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/python/autc
Instructions for updating:
Use ref() instead.
W0124 15:35:01.644590 140620629907328 deprecation.py:347] From /usr/local/lib/python:
Instructions for updating:
Use ref() instead.
WARNING:tensorflow:Skipping full serialization of Keras layer <object_detection.meta_
W0124 15:36:15.178672 140620629907328 save_impl.py:72] Skipping full serialization of
2022-01-24 15:36:46.982829: W tensorflow/python/util/util.cc:368] Sets are not currer
W0124 15:38:08.656437 140620629907328 save.py:268] Found untraced functions such as F
INFO:tensorflow:Assets written to: ./exported-models/my_model_rcnn/saved_model/assets

```

```
I0124 15:38:31.600029 140620629907328 builder_impl.py:784] Assets written to: ./export  
INFO:tensorflow:Writing pipeline config file to ./exported-models/my_model_rcnn/pipeline  
I0124 15:38:33.973511 140620629907328 config_util.py:254] Writing pipeline config file
```

• Yolo algorithm for object detection

```
yolo_path = os.path.join(capstone_path, "tensorflow/workspace/training-yolo")  
os.chdir(yolo_path)  
  
%%capture  
!git clone https://github.com/ultralytics/yolov5.git  
%cd /yolov5  
!pip install -r requirements.txt  
  
%cd ..  
  
/content/drive/My Drive/capstone/tensorflow/workspace/training-yolo  
  
%cd yolov5  
  
/content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/yolov5  
  
cd training-yolo  
  
[Errno 2] No such file or directory: 'training-yolo'  
/content/drive/My Drive/capstone/tensorflow/workspace/training-yolo  
  
ls  
  
car_data_yolo/          preprocess-yolo.py    val_csv.csv  
'Car names and make.csv' train_csv.csv      yolov5/  
  
os.chdir(capstone_path)  
  
!ls  
  
Annotations           Dataset             resnet_final_weights.h5  
arun_notebook         df_tr.csv          resnet_weights.hdf5  
aug_images_resize     df_ts.csv          tensorflow  
augmented_images.csv  labels.csv         test.csv  
'Car Images'          losses.png        tmp  
compressed_images     model_plot.png     train.csv  
compressed_test_images resnet152_weights_tf.h5 train_csv.csv  
compressed_val_images resnet_best_weights.hdf5 val_csv.csv  
dataframes            resnet_final.h5    'yolo best'
```

```
#!cp -r compressed_images yolo-path
```

```
#!cp -r compressed_val_images yolo-path
```

```
#!cp -r compressed_test_images yolo-path
```

```
dp = '/content/drive/My Drive/capstone/tensorflow/workspace/training-yolo'
dp_train = os.path.join(dp, 'compressed_images')
dp_test = os.path.join(dp, 'compressed_test_images')
dp_val = os.path.join(dp, 'compressed_val_images')
os.chdir(dp)
```

```
out_dp = '/content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/car_data_yolo'
```

```
df_train
```

	filename	width	height	class	xmin	xmax	ymin	ymax	class_name	
0	06890.jpg	1600	1200	108	196.0	1372.0	43.0	1156.0	Ford Freestar Minivan 2007	Drive/ca
1	03838.jpg	259	194	14	13.0	251.0	30.0	156.0	Audi TTS Coupe 2012	Drive/ca
2	03741.jpg	300	180	100	21.0	263.0	4.0	150.0	FIAT 500 Convertible 2012	Drive/ca
3	05159.jpg	600	450	114	9.0	600.0	66.0	373.0	Ford F-150 Regular Cab 2007	Drive/ca
4	03349.jpg	650	460	25	26.0	632.0	104.0	392.0	Audi TT RS Coupe 2012	Drive/ca
...

```
df_val
```

	filename	width	height	class	xmin	xmax	ymin	ymax	class_name
0	04506.jpg	1600	1200	63	478.0	1534.0	286.0	979.0	Chevrolet Sonic Sedan 2012
1	04600.jpg	259	194	190	17.0	241.0	47.0	158.0	Volkswagen Golf Hatchback 2012
2	04462.jpg	640	480	85	9.0	617.0	26.0	445.0	Dodge Caravan Minivan 1997

df_test_ann

	filename	width	height	class	xmin	xmax	ymin	ymax	class_name
0	00001.jpg	276	182	181	30.0	246.0	52.0	147.0	Suzuki Aerio Sedan 2007
1	00002.jpg	640	360	103	100.0	576.0	19.0	203.0	Ferrari 458 Italia Convertible 2012
2	00003.jpg	1024	741	145	51.0	968.0	105.0	659.0	Jeep Patriot SUV 2012
3	00004.jpg	640	480	187	67.0	581.0	84.0	407.0	Toyota Camry Sedan 2012
4	00005.jpg	600	373	185	140.0	593.0	151.0	339.0	Tesla Model S Sedan 2012
...

```
df_cars_name = pd.read_csv(os.path.join(dp,'Car names and make.csv'),names=['Car Name'])
df_cars_name
```

Car Name	
0	AM General Hummer SUV 2000
1	Acura RL Sedan 2012
2	Acura TL Sedan 2012
3	Acura TL Type-S 2008
4	Acura TSX Sedan 2012

```
0      AM General Hummer SUV 2000
1      Acura RL Sedan 2012
2      Acura TL Sedan 2012
3      Acura TL Type-S 2008
4      Acura TSX Sedan 2012

!ls

car_data_yolo           preprocess-yolo.py    val_csv.csv
'Car names and make.csv' train_csv.csv       yolov5
```

```
count = 0
for root, dirs, files in os.walk(dp_train):
    print(root,dirs,files)
    for file in files:
        #print(file)
        count = count + 1
        path_file = os.path.join(root,file)
        shutil.copy2(path_file,os.path.join(out_dp + '/train/images'))

print(count)

0
```

```
for root, dirs, files in os.walk(dp_test):
    for file in files:
        path_file = os.path.join(root,file)
        shutil.copy2(path_file,out_dp + '/test/images')
```

```
count = 0
for root, dirs, files in os.walk(dp_val):
    print(root,dirs,files)
    for file in files:
        count = count + 1
        path_file = os.path.join(root,file)
        shutil.copy2(path_file,os.path.join(out_dp,'val/images'))

print(count)

0
```

```
df_test = df_test_ann.copy()
```

```
df_train['filepath'] = df_train['filename'].apply(lambda x:os.path.join(out_dp + '/train/i
df_test['filepath'] = df_test['filename'].apply(lambda x:os.path.join(out_dp + '/test/imag
```

```
df_val['filepath'] = df_val['filename'].apply(lambda x:os.path.join(out_dp + '/val/images' + x))

l = list(df_train['filepath'])
print(l[0])

/content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/car_data_yolo/trai

l = list(df_test['filepath'])
print(l[0])

/content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/car_data_yolo/test

l = list(df_val['filepath'])
print(l[0])

/content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/car_data_yolo/val

df_train['Label_Path'] = df_train['filename'].apply(lambda x:os.path.join(out_dp + '/train/label' + x))
df_test['Label_Path'] = df_test['filename'].apply(lambda x:os.path.join(out_dp + '/test/label' + x))
df_val['Label_Path'] = df_val['filename'].apply(lambda x:os.path.join(out_dp + '/val/label' + x))

l = list(df_val['Label_Path'])
print(l[0])

/content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/car_data_yolo/val

l = list(df_test['Label_Path'])
print(l[0])

/content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/car_data_yolo/test

l = list(df_train['Label_Path'])
print(l[0])

/content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/car_data_yolo/trai

for index, row in df_train.iterrows():
    x, y, w, h = YOLO_xywh(row['xmin'],row['ymin'],row['xmax'],row['ymax'],row['width'], row['height'])
    image_class_index = int(row['class']) - 1
    print(row['Label_Path'])
    with open(row['Label_Path'], "w") as text_file:
        print(f"{image_class_index} {x} {y} {w} {h}", file=text_file)

/content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/car_data_yolo/trai
```

```
for index, row in df_val.iterrows():
```

```
x, y, w, h = YOLO_xywh(row['xmin'], row['ymin'], row['xmax'], row['ymax'], row['width'],
image_class_index = int(row['class']) - 1
print(row['Label_Path'])
with open(row['Label_Path'], "w") as text_file:
    print(f"{image_class_index} {x} {y} {w} {h}", file=text_file)

for index, row in df_test.iterrows():
    x, y, w, h = YOLO_xywh(row['xmin'],
                           row['ymin'],
                           row['xmax'],
                           row['ymax'],
                           row['width'],
                           row['height'])
    image_class_index = int(row['class']) - 1
    print(row['Label_Path'])
    with open(row['Label_Path'], "w") as text_file:
        print(f"{image_class_index} {x} {y} {w} {h}", file=text_file)
```

```
%cd yolov5
```

```
/content/drive/My Drive/capstone/tensorflow/workspace/training-yolo/yolov5
```

```
%cd data
```

```
/content/drive/My Drive/capstone/tensorflow/workspace/training-yolo/yolov5/data
```

```
!touch carData.yaml
```

```
%%writefile carData.yaml
```

```
train: /content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/car_data_yolo/train
val: /content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/car_data_yolo/val
test: /content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/car_data_yolo/test
```

```
# number of classes
```

```
nc: 196
```

```
# class names
```

```
names: ['AM General Hummer SUV 2000', 'Acura RL Sedan 2012', 'Acura TL Sedan 2012', 'Acura
```

```
%cat carData.yaml
```

```
train: /content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/car_data_yolo/train
val: /content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/car_data_yolo/val
test: /content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/car_data_yolo/test
```

```
# number of classes
```

```
nc: 196
```

```
# class names
```

```
names: ['AM General Hummer SUV 2000', 'Acura RL Sedan 2012', 'Acura TL Sedan 2012',
```

```
%cd /content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/yolov5
```

```
/content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/yolov5
```

```
!python train.py --img 224 --cfg yolov5s.yaml --hyp hyp.scratch.yaml --batch 32 --epochs 2
```

Epoch	gpu_mem	box	obj	cls	labels	img_size
105/199	1.73G	0.01768	0.007875	0.05542	53	224: 100% 204/2
	Class	Images	Labels		P	R mAP@.5 mAP@.5
	all	1629	1629	0.627	0.768	0.776 ε
Epoch	gpu_mem	box	obj	cls	labels	img_size
106/199	1.73G	0.0176	0.007773	0.0545	52	224: 100% 204/2
	Class	Images	Labels		P	R mAP@.5 mAP@.5
	all	1629	1629	0.634	0.774	0.784 ε
Epoch	gpu_mem	box	obj	cls	labels	img_size
107/199	1.73G	0.01756	0.007809	0.05455	52	224: 100% 204/2
	Class	Images	Labels		P	R mAP@.5 mAP@.5
	all	1629	1629	0.614	0.801	0.787 ε
Epoch	gpu_mem	box	obj	cls	labels	img_size
108/199	1.73G	0.01767	0.007878	0.05348	52	224: 100% 204/2
	Class	Images	Labels		P	R mAP@.5 mAP@.5
	all	1629	1629	0.63	0.792	0.791 ε
Epoch	gpu_mem	box	obj	cls	labels	img_size
109/199	1.73G	0.01782	0.007859	0.05266	58	224: 100% 204/2
	Class	Images	Labels		P	R mAP@.5 mAP@.5
	all	1629	1629	0.627	0.795	0.795 ε
Epoch	gpu_mem	box	obj	cls	labels	img_size
110/199	1.73G	0.01767	0.007814	0.05219	50	224: 100% 204/2
	Class	Images	Labels		P	R mAP@.5 mAP@.5
	all	1629	1629	0.659	0.776	0.799 ε
Epoch	gpu_mem	box	obj	cls	labels	img_size
111/199	1.73G	0.01759	0.007875	0.05273	58	224: 100% 204/2
	Class	Images	Labels		P	R mAP@.5 mAP@.5
	all	1629	1629	0.686	0.769	0.804 ε
Epoch	gpu_mem	box	obj	cls	labels	img_size
112/199	1.73G	0.01767	0.007867	0.05193	49	224: 100% 204/2
	Class	Images	Labels		P	R mAP@.5 mAP@.5
	all	1629	1629	0.707	0.763	0.807 ε
Epoch	gpu_mem	box	obj	cls	labels	img_size
113/199	1.73G	0.01761	0.007742	0.05144	39	224: 100% 204/2
	Class	Images	Labels		P	R mAP@.5 mAP@.5
	all	1629	1629	0.715	0.766	0.809 ε
Epoch	gpu_mem	box	obj	cls	labels	img_size
114/199	1.73G	0.01747	0.00782	0.05148	42	224: 100% 204/2
	Class	Images	Labels		P	R mAP@.5 mAP@.5
	all	1629	1629	0.724	0.769	0.812 ε
Epoch	gpu_mem	box	obj	cls	labels	img_size
115/199	1.73G	0.01750	0.007877	0.05016	50	224: 100% 204/2

1/30/22, 6:58 PM

capstone_cv2_final.ipynb - Colaboratory							
	Class	Images	Labels	P	R	mAP@.5	mAP@.5
	all	1629	1629	0.695	0.787	0.815	0
Epoch	gpu_mem	box	obj	cls	labels	img_size	
116/199	1.73G	0.01728	0.00778	0.05056	52	224: 100% 204/2	

!ls

sample_data

!python val.py --img 224 --weights /content/drive/MyDrive/capstone/tensorflow/workspace/tr

val: data=/content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/yolov5/c
YOLOv5 🚀 v6.0-200-g9708cf5 torch 1.10.0+cu111 CUDA:0 (Tesla P100-PCIE-16GB, 16281Mi

Fusing layers...

Model Summary: 213 layers, 7538737 parameters, 0 gradients

test: Scanning '/content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/c
Class Images Labels P R mAP@.5 mAP@.5:
all 8040 8040 0.789 0.86 0.881 0.86

Speed: 0.0ms pre-process, 1.0ms inference, 1.0ms NMS per image at shape (32, 3, 224,
Results saved to runs/val/yolo_car_class_det5

!python export.py --weights runs/train/yolo_car_classification6/weights/best.pt --img 224

export: data=data/coco128.yaml, weights=['runs/train/yolo_car_classification6/weight
YOLOv5 🚀 v6.0-200-g9708cf5 torch 1.10.0+cu111 CPU

Fusing layers...

Model Summary: 213 layers, 7538737 parameters, 0 gradients

PyTorch: starting from runs/train/yolo_car_classification6/weights/best.pt (60.9 MB)

TorchScript: starting export with torch 1.10.0+cu111...

/content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/yolov5/models/yolo
if self.onnx_dynamic or self.grid[i].shape[2:4] != x[i].shape[2:4]:

/content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/yolov5/models/yolo
if self.onnx_dynamic or self.grid[i].shape[2:4] != x[i].shape[2:4]:

TorchScript: export success, saved as runs/train/yolo_car_classification6/weights/bes

requirements: onnx not found and is required by YOLOv5, attempting auto-update...

Collecting onnx

Downloading onnx-1.10.2-cp37-cp37m-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (

Requirement already satisfied: numpy>=1.16.6 in /usr/local/lib/python3.7/dist-packages

Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from onnx)

Requirement already satisfied: protobuf in /usr/local/lib/python3.7/dist-packages (from onnx)

Requirement already satisfied: typing-extensions>=3.6.2.1 in /usr/local/lib/python3.7/dist-pac

Installing collected packages: onnx

Successfully installed onnx-1.10.2

requirements: 1 package updated per ['onnx']

requirements: ⚠ Restart runtime or rerun command for updates to take effect

ONNX: starting export with onnx 1.10.2...

```
/content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/yolov5/models/yolo
  if self.onnx_dynamic or self.grid[i].shape[2:4] != x[i].shape[2:4]:
ONNX: export success, saved as runs/train/yolo_car_classification6/weights/best.onnx

Export complete (8.51s)
Results saved to /content/drive/MyDrive/capstone/tensorflow/workspace/training-yolo/
Visualize with https://netron.app
Detect with `python detect.py --weights runs/train/yolo_car_classification6/weights/best.onnx
Validate with `python val.py --weights runs/train/yolo_car_classification6/weights/best.onnx`
```



▼ Object detection with different models:

```
IMAGE_PATHS = list(df_test_ann['filepath'])
IMAGE_PATHS
['/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Chrysler Sebring',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/BMW ActiveHybrid',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Hyundai Veracruz',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Bentley Mulsanne',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Audi 100 Sedan 1',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Mercedes-Benz E-',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Dodge Dakota Cre',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Chevrolet Camaro',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Chevrolet Silver',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Volvo 240 Sedan',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Chevrolet Cobalt',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Chevrolet Monte',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Audi R8 Coupe 20',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Mercedes-Benz E-',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Ram C-V Cargo Va',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Volkswagen Beetl',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Acura TL Type-S',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Ferrari 458 Ital',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Cadillac Escalad',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Honda Odyssey Mi',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Toyota Corolla S',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Honda Odyssey Mi',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/BMW ActiveHybrid',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/FIAT 500 Abarth',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Dodge Durango SU',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/BMW 3 Series Sed',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Mercedes-Benz C-',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/McLaren MP4-12C',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Chevrolet Corvet',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Chevrolet Silver',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Rolls-Royce Phan',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Acura ZDX Hatchb',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Volkswagen Golf',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/BMW M5 Sedan 201',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Aston Martin Vir',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Mitsubishi Lance',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Chevrolet Corvet',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Hyundai Veracruz',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Dodge Journey SU',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Chrysler Town an',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Chevrolet Cobalt',
 '/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Chevrolet Cobalt']
```

```
/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Honda Odyssey Mi  
'/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Honda Odyssey Mi  
'/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Toyota 4Runner S  
'/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Fisker Karma Sed  
'/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Acura ZDX Hatchb  
'/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Spyker C8 Conver  
'/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Chevrolet Sonic  
'/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Dodge Caliber Wa  
'/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Chevrolet Impala  
'/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Ferrari 458 Ital  
'/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Audi TTS Coupe 2  
'/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Maybach Landaule  
'/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Ford F-150 Regul
```

```
'/content/drive/My Drive/capstone/Dataset/Car Images/Test Images/Toyota Camry Sed
```



```
os.getcwd()
```

```
'/content/drive/MyDrive/capstone/tensorflow/models/research'
```

```
%cd ..
```

```
/content/drive/MyDrive/capstone/tensorflow
```

```
%cd workspace/training-ssd/exported-models/my_model/
```

```
/content/drive/MyDrive/capstone/tensorflow/workspace/training-ssd/exported-models/my_
```



```
#PATH_TO_MODEL_DIR = "workspace/training-ssd/exported-models/my_model"
```

```
#os.chdir(PATH_TO_MODEL_DIR)
```

```
import time
```

```
from object_detection.utils import label_map_util
```

```
from object_detection.utils import visualization_utils as viz_utils
```

```
#PATH_TO_SAVED_MODEL = PATH_TO_MODEL_DIR + "saved_model"
```

```
print('Loading model...', end='')
```

```
start_time = time.time()
```

```
# Load saved model and build the detection function
```

```
detect_fn = tf.saved_model.load("saved_model")
```

```
end_time = time.time()
```

```
elapsed_time = end_time - start_time
```

```
print('Done! Took {} seconds'.format(elapsed_time))
```

```
Loading model...Done! Took 23.845096588134766 seconds
```

```
def download_labels(filename):
```

```
    #base_url = 'https://raw.githubusercontent.com/tensorflow/models/master/research/objec
```

```
https://colab.research.google.com/drive/1ioNccQebSjx__adl3gJgG_OYAlldUb1GZ#scrollTo=7kPe9jIP5knA&printMode=true
```

```
38/46
```

```
#label_dir = tf.keras.utils.get_file(fname=filename,
#                                     origin=base_url + filename,
#                                     untar=False)
label_dir = os.path.join("/content/drive/MyDrive/capstone/tensorflow/workspace/training-ssd/annotations/label_map.pbtxt")
return str(label_dir)

LABEL_FILENAME = 'label_map.pbtxt'
PATH_TO_LABELS = download_labels(LABEL_FILENAME)

PATH_TO_LABELS

'/content/drive/MyDrive/capstone/tensorflow/workspace/training-ssd/annotations/label_map.pbtxt'

category_index = label_map_util.create_category_index_from_labelmap(PATH_TO_LABELS,
                                                                use_display_name=True)

category_index
```

140: {'id': 140, 'name': 'Hyundai Azera Sedan 2012'},
141: {'id': 141, 'name': 'Infiniti G Coupe IPL 2012'},
142: {'id': 142, 'name': 'Infiniti QX56 SUV 2011'},
143: {'id': 143, 'name': 'Isuzu Ascender SUV 2008'},
144: {'id': 144, 'name': 'Jaguar XK XKR 2012'},
145: {'id': 145, 'name': 'Jeep Patriot SUV 2012'},
146: {'id': 146, 'name': 'Jeep Wrangler SUV 2012'},
147: {'id': 147, 'name': 'Jeep Liberty SUV 2012'},
148: {'id': 148, 'name': 'Jeep Grand Cherokee SUV 2012'},
149: {'id': 149, 'name': 'Jeep Compass SUV 2012'},
150: {'id': 150, 'name': 'Lamborghini Reventon Coupe 2008'},
151: {'id': 151, 'name': 'Lamborghini Aventador Coupe 2012'},
152: {'id': 152, 'name': 'Lamborghini Gallardo LP 570-4 Superleggera 2012'},
153: {'id': 153, 'name': 'Lamborghini Diablo Coupe 2001'},
154: {'id': 154, 'name': 'Land Rover Range Rover SUV 2012'},
155: {'id': 155, 'name': 'Land Rover LR2 SUV 2012'},
156: {'id': 156, 'name': 'Lincoln Town Car Sedan 2011'},
157: {'id': 157, 'name': 'MINI Cooper Roadster Convertible 2012'},
158: {'id': 158, 'name': 'Maybach Landaulet Convertible 2012'},
159: {'id': 159, 'name': 'Mazda Tribute SUV 2011'},
160: {'id': 160, 'name': 'McLaren MP4-12C Coupe 2012'},
161: {'id': 161, 'name': 'Mercedes-Benz 300-Class Convertible 1993'},
162: {'id': 162, 'name': 'Mercedes-Benz C-Class Sedan 2012'},
163: {'id': 163, 'name': 'Mercedes-Benz SL-Class Coupe 2009'},
164: {'id': 164, 'name': 'Mercedes-Benz E-Class Sedan 2012'},
165: {'id': 165, 'name': 'Mercedes-Benz S-Class Sedan 2012'},
166: {'id': 166, 'name': 'Mercedes-Benz Sprinter Van 2012'},
167: {'id': 167, 'name': 'Mitsubishi Lancer Sedan 2012'},
168: {'id': 168, 'name': 'Nissan Leaf Hatchback 2012'},
169: {'id': 169, 'name': 'Nissan NV Passenger Van 2012'},
170: {'id': 170, 'name': 'Nissan Juke Hatchback 2012'},
171: {'id': 171, 'name': 'Nissan 240SX Coupe 1998'},
172: {'id': 172, 'name': 'Plymouth Neon Coupe 1999'},
173: {'id': 173, 'name': 'Porsche Panamera Sedan 2012'},
174: {'id': 174, 'name': 'Ram C-V Cargo Van Minivan 2012'},
175: {'id': 175, 'name': 'Rolls-Royce Phantom Drophead Coupe Convertible 2012'},
176: {'id': 176, 'name': 'Rolls-Royce Ghost Sedan 2012'},
177: {'id': 177, 'name': 'Rolls-Royce Phantom Sedan 2012'}

```
177: {'id': 177, 'name': 'Rolls-Royce Phantom Sedan 2012'},
178: {'id': 178, 'name': 'Scion xD Hatchback 2012'},
179: {'id': 179, 'name': 'Spyker C8 Convertible 2009'},
180: {'id': 180, 'name': 'Spyker C8 Coupe 2009'},
181: {'id': 181, 'name': 'Suzuki Aerio Sedan 2007'},
182: {'id': 182, 'name': 'Suzuki Kizashi Sedan 2012'},
183: {'id': 183, 'name': 'Suzuki SX4 Hatchback 2012'},
184: {'id': 184, 'name': 'Suzuki SX4 Sedan 2012'},
185: {'id': 185, 'name': 'Tesla Model S Sedan 2012'},
186: {'id': 186, 'name': 'Toyota Sequoia SUV 2012'},
187: {'id': 187, 'name': 'Toyota Camry Sedan 2012'},
188: {'id': 188, 'name': 'Toyota Corolla Sedan 2012'},
189: {'id': 189, 'name': 'Toyota 4Runner SUV 2012'},
190: {'id': 190, 'name': 'Volkswagen Golf Hatchback 2012'},
191: {'id': 191, 'name': 'Volkswagen Golf Hatchback 1991'},
192: {'id': 192, 'name': 'Volkswagen Beetle Hatchback 2012'},
193: {'id': 193, 'name': 'Volvo C30 Hatchback 2012'},
194: {'id': 194, 'name': 'Volvo 240 Sedan 1993'},
195: {'id': 195, 'name': 'Volvo XC90 SUV 2007'},
196: {'id': 196, 'name': 'smart fortwo Convertible 2012'}}
```

```
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore') # Suppress Matplotlib warnings
```

```
def load_image_into_numpy_array(path):
    """Load an image from file into a numpy array.
```

Puts image into numpy array to feed into tensorflow graph.
Note that by convention we put it into a numpy array with shape
(height, width, channels), where channels=3 for RGB.

Args:

path: the file path to the image

Returns:

uint8 numpy array with shape (img_height, img_width, 3)

"""

```
return np.array(Image.open(path))
```

```
for image_path in IMAGE_PATHS:
```

```
    print('Running inference for {}... '.format(image_path), end='')
```

```
    image_np = load_image_into_numpy_array(image_path)
```

```
    # Things to try:
    # Flip horizontally
    # image_np = np.fliplr(image_np).copy()
```

```
    # Convert image to grayscale
    # image_np = np.tile(
    #     np.mean(image_np, 2, keepdims=True), (1, 1, 3)).astype(np.uint8)
```

```
# The input needs to be a tensor, convert it using `tf.convert_to_tensor`.
input_tensor = tf.convert_to_tensor(image_np)
# The model expects a batch of images, so add an axis with `tf.newaxis`.
input_tensor = input_tensor[tf.newaxis, ...]

# input_tensor = np.expand_dims(image_np, 0)
detections = detect_fn(input_tensor)

# All outputs are batches tensors.
# Convert to numpy arrays, and take index [0] to remove the batch dimension.
# We're only interested in the first num_detections.
num_detections = int(detections.pop('num_detections'))
detections = {key: value[0, :num_detections].numpy()
               for key, value in detections.items()}
detections['num_detections'] = num_detections

# detection_classes should be ints.
detections['detection_classes'] = detections['detection_classes'].astype(np.int64)

image_np_with_detections = image_np.copy()

viz_utils.visualize_boxes_and_labels_on_image_array(
    image_np_with_detections,
    detections['detection_boxes'],
    detections['detection_classes'],
    detections['detection_scores'],
    category_index,
    use_normalized_coordinates=True,
    max_boxes_to_draw=200,
    min_score_thresh=.30,
    agnostic_mode=False)

plt.figure()
plt.imshow(image_np_with_detections)
print('Done')
plt.show()
```

```
ValueError                                Traceback (most recent call last)
<ipython-input-70-5c2ce9f3fedc> in <module>()
      41
      42     # input_tensor = np.expand_dims(image_np, 0)
---> 43     detections = detect_fn(input_tensor)
      44
      45     # All outputs are batches tensors.
```

◆ 2 frames

```
/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/function.py in
_convert_inputs_to_signature(inputs, input_signature, flat_input_signature)
 2980     flat_input_signature,
 2981     flatten_inputs)):
-> 2982     raise ValueError("Python inputs incompatible with input_signature:\n"
 2983                      f"{format_error_message(inputs, input_signature)}")
 2984
```

ValueError: Python inputs incompatible with input signature:

```
inputs: (  
    tf.Tensor(  
[[[ 87  87  87 ... 151 151 151]  
 [ 86  85  85 ... 152 152 152]  
 [ 82  82  82 ... 152 152 152]  
 ...  
 [123 123 123 ... 123 123 123]
```

```
[133 133 133 ... 133 133 133]  
[133 133 133 ... 133 133 133]]], shape=(1, 400, 600), dtype=uint8))  
input_signature: (  
    TensorSpec(shape=(1, None, None, 3), dtype=tf.uint8, name='input_tensor')).
```

SEARCH STACK OVERFLOW

