# Source Code

# Install necessary packages

3 !pip install ultralytics

4 !pip install −U ray[tune]

5 !pip install −U ipywidgets

6

7 # Import libraries

8 import warnings

9 warnings . filterwarnings ( ’ignore ’)

10

11 import os

12 import shutil

13 import numpy as np

14 import pandas as pd

15 import matplotlib . pyplot as plt

16 import seaborn as sns

17 import cv2

18 import yaml

19 from PIL import Image

20 from ultralytics import YOLO

21 from IPython . display import Video

22

23 print (”Import Completed Successfully”)

24

25 # Configure seaborn

26 sns . set ( rc={’axes . facecolor ’ : ’#eae8fa ’}, style=’darkgrid ’)

27 print (”Setup Completed Successfully”)

28

29 # Load pretrained YOLOv8 model

30 model = YOLO( ’yolov8n . pt ’)

31

32 # Perform prediction on sample image

33 image path = ’/ content / sample image . jpg ’

34 results = model. predict ( source=image path , imgsz=640, conf =0.5)

35 sample image = results [0]. plot ( line width =2)

36 sample image = cv2. cvtColor (sample image , cv2.COLOR BGR2RGB)

37

38 plt . figure ( figsize =(20 ,15) )

plt . imshow( sample image )

40 plt . t i t l e ( ’Detected Objects in Sample Image by YOLOv8’ , fontsize =20)

41 plt . axis ( ’ off ’)

42 plt . show()

43

44 # Load and display dataset information

45 dataset path = ’/ content / drive /MyDrive/Major−Project (Research−papers ) /

Vehicle Detection Image Dataset ’

46 yaml file path = os. path . join ( dataset path , ’data .yaml’)

47

48 with open( yaml file path , ’r ’) as file :

49

50

51

yaml content = yaml. load ( file , Loader=yaml. FullLoader )

print (yaml .dump( yaml content , default flow

52 # Count training and validation images

53 train

s t yle=False ) )

i mages path = os. path . join ( dataset path , ’ train ’ , ’images ’)

54 valid images path = os. path . join ( dataset path , ’valid ’ , ’images ’)

55

56 num train images = sum(1 for file in os. listdir ( train

i mages path ) if file . endswith( ’. jpg ’) )

57 num valid images = sum(1 for file in os. listdir ( valid images path ) if file . endswith( ’. jpg ’))

58

59 print ( f”Number of training images : {num train images}”)

60 print ( f”Number of validation images : {num valid images}”)

61

62 # Display sample training images

63 image files = [ file for file in os. listdir ( train

i mages path ) if file . endswith( ’. jpg ’) ]

64 selected images = image files [::max(1 , len ( image files ) // 8)]

65

66 fig , axes = plt . subplots (2 , 4, figsize =(20, 11))

67 for ax , img

68

f i l e in zip (axes . ravel () , selected images ) :

i mg path = os. path . join ( train images path , img

69

70

71

image = Image.open( img path)

ax . imshow(image)

ax . axis ( ’ off ’)

f i l e )

72 plt . suptitle ( ’Sample Images from Training Dataset ’ , fontsize =20)

73 plt . tight layout ()

74 plt . show()

75

76 # Load trained model

77 best model path = ’/ content / runs / detect / train / weights / best . pt ’

78 best model = YOLO( best model path )

79

80 # Evaluate model performance

81 metrics = best model . val ( split=’val ’)

82 metrics df = pd.DataFrame . from dict ( metrics . results dict , orient=’index ’ , columns=[ ’Metric Value ’ ])

83 metrics df . round (3)

84

85 # Display predictions on validation set

86 image files = [ file for file in os. listdir ( valid images path ) if file . endswith( ’. jpg ’)]

87 selected images = image files [::max(1 , len ( image files ) // 9)]

fig , axes = plt . subplots (3 , 3, figsize =(20, 21))

90 fig . suptitle ( ’ Validation Set Inferences ’ , fontsize =24)

91 for i , ax in enumerate(axes . flatten () ) :

92

93

94

95

96

97

i mage path = os. path . join ( valid images path , selected images [ i ])

r e sults = best model . predict ( source=image path , imgsz=640, conf =0.5)

annotated image = results [0]. plot ( line width =1)

annotated image rgb = cv2. cvtColor ( annotated image , cv2.COLOR BGR2RGB)

ax . imshow( annotated image rgb )

ax . axis ( ’ off ’)

98 plt . tight layout ()

99 plt . show()

100

101 # Analyze traffic density from video

102 sample video = ’/ sample video .mp4’

103 dataset video

path = ’/ content / drive /MyDrive/Major−Project (Research−papers ) /

Vehicle Detection Image Dataset / sample video .mp4’

104 shutil . copyfile ( dataset video

path , sample video )

105 best model . predict ( source=sample video , save=True)

106

107 ! ffmpeg −y −loglevel panic −i / content / runs / detect / predict / sample video . avi processed sample video .

mp4

108 Video (” processed sample video .mp4” , embed=True , width=960)

109

110 # Traffic region definition

111 vertices1 = np. array ([(465 , 350) , (609, 350) , (510, 630) , (2 , 630)] , dtype=np. int32 )

112 vertices2 = np. array ([(678 , 350) , (815, 350) , (1203, 630) , (743, 630)] , dtype=np. int32 )

113

114 # Traffic analysis

115 cap = cv2. VideoCapture ( sample video )

116 out = cv2. VideoWriter ( ’/ content / traffic

20.0 , (1280, 720))

117

118 while cap . isOpened () :

119

120

121

122

123

124

125

r et , frame = cap. read ()

i f not ret :

break

detection frame = frame .copy ()

detection frame [:325 , :] = 0

detection frame [635: , :] = 0

d ensity

a nalysis . avi ’ , cv2 . VideoWriter fourcc (\* ’XVID’) ,

r e sults = best model . predict ( detection frame , imgsz=640, conf =0.4)

126

127

128

129

130

131

132

133

134

processed frame = results [0]. plot ( line width =1)

processed frame [:325 , :] = frame[:325 , :]

processed frame [635: , :] = frame[635: , :]

bounding boxes = results [0]. boxes

l eft , right = 0, 0

f or box in bounding boxes .xyxy:

i f box[0] < 609:

l e f t += 1

else :

136

137

138

139

140

141

142

143

144

145

r i ght += 1

cv2 . polylines ( processed frame , [ vertices1 ] , True , (0 , 255, 0) , 2)

cv2 . polylines ( processed frame , [ vertices2 ] , True , (255 , 0, 0) , 2)

cv2 . putText ( processed frame , f”Left Lane: {left} − {’Heavy’ if left > 10 else ’Smooth’}” , (10,

50) ,

cv2 .FONT HERSHEY SIMPLEX, 1, (255, 255, 255) , 2)

cv2 . putText ( processed frame , f”Right Lane: {right} − {’Heavy’ if right > 10 else ’Smooth’}” ,

(820 , 50) ,

cv2 .FONT HERSHEY SIMPLEX, 1, (255, 255, 255) , 2)

out . write ( processed frame )

146 cap . release ()

147 out . release ()

148

149 ! ffmpeg −y −loglevel panic −i / content / traffic

150 Video (” traffic

d ensity

d ensity

a nalysis . avi traffic

a nalysis .mp4” , embed=True , width=960)

151

152 # Export model to ONNX

153 best model . export ( format=’onnx ’)