TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
KHOA ĐIỆN – ĐIỆN TỬ
----------o0o----------

BÀI TẬP LỚN MÔN HỌC

XỬ LÍ ẢNH TRONG CÔNG NGHIỆP VÀ GIAO THÔNG

Đề 9

Ứng dụng xử lý ảnh để đếm và phân loại phương tiện giao thông

Sinh viên thực hiện: Dương Xuân Ngọc
Lớp KTĐKTĐH3 - K62
Nhóm 6

Giảng viên hướng dẫn: TS. Phí Văn Lâm

Hà Nội, tháng 12 năm 2023

MỤC LỤC

Phần 1: Giới thiệu đề tài và mục tiều đề ra.

1.1. Giới thiệu đề tài.

Đề tài "Ứng dụng xử lý ảnh để đếm và phân loại phương tiện giao thông" tập trung vào việc phát triển hệ thống thông minh nhằm nhận diện, đếm, và phân loại các phương tiện giao thông trong môi trường đô thị. Chúng ta có thể sử dụng mô hình YOLO và các thuật toán xử lý ảnh, nghiên cứu này hứa hẹn cung cấp giải pháp hiệu quả cho quản lý lưu lượng giao thông và giám sát an toàn đường phố.

1.2. Mục tiêu.

Đề tài nhằm đặt ra và đạt được những mục tiêu chính sau:

- Nhận diện chính xác: phát triển mô hình nhận diện đối tượng chính xác, ở đây chúng ta xe nhận diện phân loại 3 loại phương tiện giao thông chính ô tô, xe máy và xe đạp.
- Đếm lưu lượng giao thông: Xây dựng thuật toán đếm lưu lượng giao thông hiệu quả, giúp đánh giá và theo dõi sự di chuyển của các phương tiện trong thời gian thực.
- ứng dụng thực tế: Kiểm nghiệm và triển khai giải pháp trong các tình huống thực tế để đảm bảo tính khả thi và hiệu quả của hệ thống trong môi trường đô thị.
- Đóng góp cho xã hội: Cung cấp giải pháp hữu ích và tiên tiến cho quản lý giao thông đô thị, góp phần vào việc xây dựng đô thị thông minh, an toàn và hiệu quả.

Phần 2: Quy trình thực hiện.

2.1. Lựa chọn ngôn ngữ và môi trường lập trình.

Có nhiều ngôn ngữ lập trình được sử dụng để xử lý ảnh, tuy nhiên, một số ngôn ngữ phổ biến nhất trong lĩnh vực này là:

- Python: Python là một ngôn ngữ lập trình phổ biến trong xử lý ảnh, được sử dụng cho các thư viện như OpenCV, Pillow và scikit-image.
- MATLAB: MATLAB cũng là một ngôn ngữ phổ biến trong lĩnh vực xử lý ảnh. Nó cung cấp các công cụ và thư viện xử lý ảnh cho các nhà khoa học và kỹ sư.
- C++: C++ là một ngôn ngữ lập trình mạnh mẽ và nhanh chóng, được sử dụng rộng rãi trong lĩnh vực xử lý ảnh. OpenCV là một thư viện phổ biến được sử dụng để xử lý ảnh bằng C++.
- Java: Java cũng được sử dụng trong lĩnh vực xử lý ảnh, đặc biệt là trong các ứng dụng di động. Các thư viện như JavaCV và OpenCV for Java cung cấp các công cụ và khung làm việc cho việc xử lý ảnh trong Java.

Để thực hiện giải quyết bài toán trên lựa chọn ngôn ngữ lập trình Python Với môi trường lập trình cũng có rất nhiều môi trường phổ biến để hỗ trợ lập trình python như Pycharm, Sublime Text, VSC,.... ở đây chúng ta sẽ sử dụng VSC đề làm môi trường lập trình.

2.2. Nhận diện đối tượng bằng mô hình học máy.

Các thuật toán nhận dạng đối tượng là một lĩnh vực phát triển mạnh mẽ trong khoa học máy tính. Dưới đây là một số thuật toán phổ biến:

- các mô hình RCNN (RCNN, Fast RCNN, Faster - RCNN,…) : điểm nổi bật là được sử dụng với độ chính xác cao, có yêu cầu cao về phân loại, vị trí của đối tượng, vật thể. Điểm nổi bật có thể kể đến là hầu hết các thuật toán này đều đem lại độ chính xác cực kì cao, bù lại cho khả năng "realtime" thấp nên được sử dụng cho các bài toán không cần hoặc không yêu cầu lớn về thời gian nhận diện hình ảnh, vật thể...
- Nhóm các mô hình theo họ Yolo: về nhóm thuật toán này là khả năng nhận diện và xử lí nhanh, hầu hết nhóm này được đánh giá cao về khả năng "realtime" trong nhận diện vật thể. Có thể xem như đây là nhóm thuật toán có ứng dụng cao trong các ứng dụng cần thời gian xử lí nhanh với yêu cầu độ chính xác ở mức "tương đối". Tuy vậy, hiện nay các nhóm tác giả đã liên tục ra các phiên bản nâng cấp của nhóm thuật toán để cải thiện về độ chính xác và tốc độ nhân diện..

Trong đó Yolo là một mô hình mạng CNN cho việc phát hiện, nhận dạng, phân loại đối tượng. Yolo được tạo ra từ việc kết hợp giữa các convolutional layers và connected layers.Trong đó các convolutional layers sẽ trích xuất ra các feature của ảnh, còn full-connected layers sẽ dự đoán ra xác suất đó và tọa độ của đối tượng.hiện 11 và nhận dạng trong ảnh và video. Vì vập bài toán trên sẽ sử dụng thuật toán này để hỗ trợ cho quá trình phân loại và đếm số xe.

2.3.Thư viện sử dụng.

Để triển khai đề tài "Ứng dụng xử lý ảnh để đếm và phân loại phương tiện giao thông," chúng ta sử dụng các thư viện và công cụ mạnh mẽ trong ngôn ngữ lập trình Python. Dưới đây các thư viện được sử dụng:

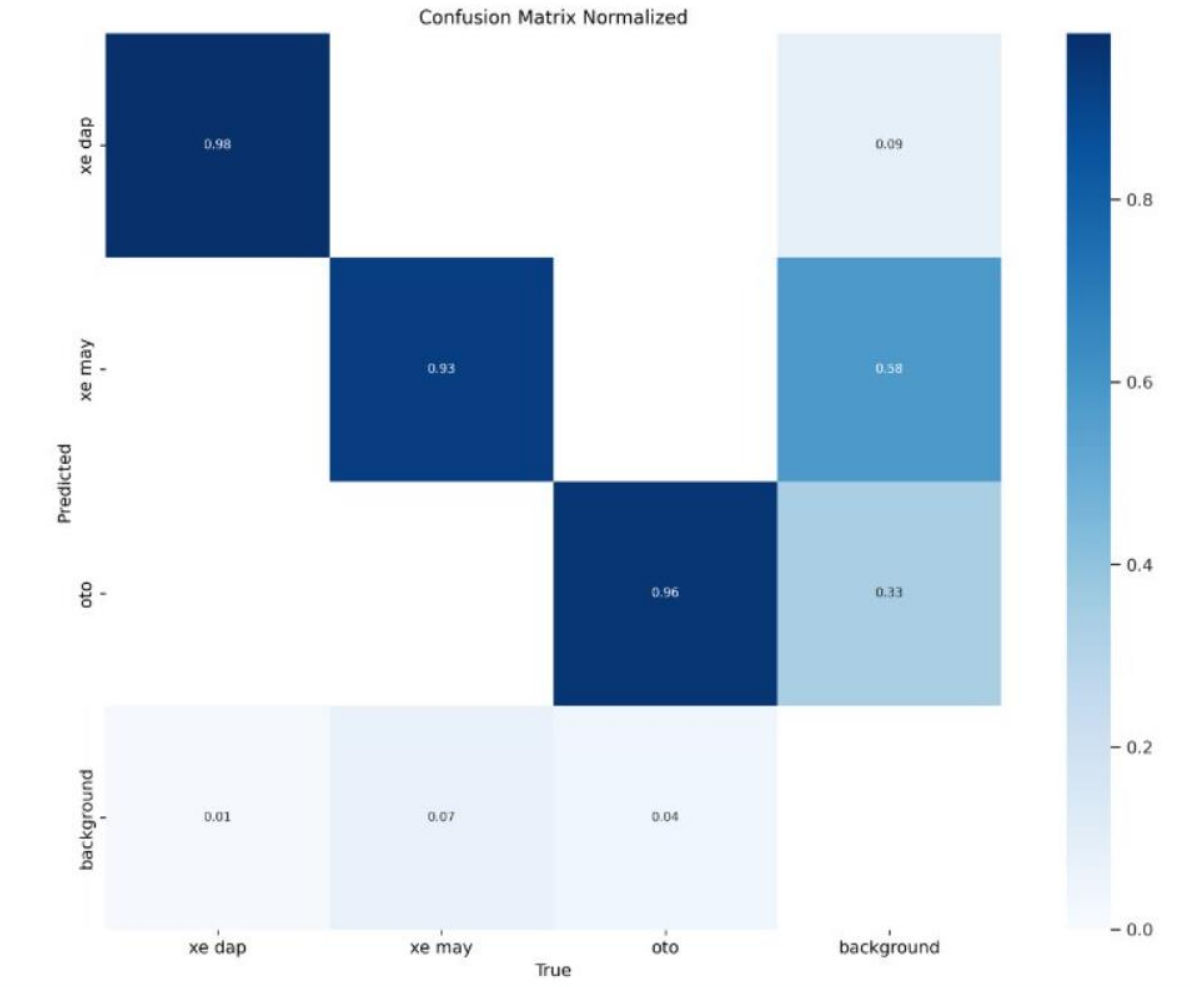
- “from ultralytics import YOLO”: thư viện từ ultralytics để nhận diện đối tượng.
- “import cv2”: thư viện xử lí ảnh và video.
- “import cvzone”: thư viện toán học cho các phép tính toán cơ bản.
- “from sort import *”: thư viện theo dõi đối tượng.
- “from datetime import datetime”: thư viện để xử lý thời gian
- “ from openpyxl import workbook; from openpyxl.styles import Font”: thư viện để làm việc với tệp excel.

2.4. chọn model để xử lí nhận diện.

Ta sẽ sử dụng phiên bản mới nhất là yolov8. Để phù hợp cũng như tăng độ chính cho bài toán chúng ta sử dụng yolov8x.

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

- Kết quả sau khi huấn luyện model:
    - Bảng confusion matrix



Phần 3:Thuật Toán.

3.1.Thuật toán YOLO.

Để phân biệt xe đạp, xe máy, ô tô trong ứng dụng xử lý ảnh, chúng ta sử dụng mô hình YOLO (You Only Look Once), một mô hình nhận diện đối tượng hiệu quả và nhanh chóng. Dưới đây là quy trình và phương pháp chi tiết:

- Chuẩn bị dữ liệu:

- Thu thập và chuẩn bị bộ dữ liệu chứa ảnh có chứa các phương tiện xe đạp, xe máy, ô tô.
- Gán nhãn cho từng ảnh để đào tạo mô hình với thông tin vị trí và loại của các đối tượng.

- Chọn mô hình yolo:

- Chọn phiên bản của mô hình YOLO, ở đây chúng ta sử dụng mô hình yolov8x một phiên bản mới nhất hiện nay với độ nhận diện và tốc độ xử lí nhanh chóng.

- Đào tạo mô hình

- Sử dụng bộ dữ liệu đã chuẩn bị để đào tạo mô hình YOLO. Quá trình này giúp mô hình học cách nhận diện và phân loại các phương tiện từ dữ liệu hình ảnh.

- Tinh chỉnh tham số

- Tinh chỉnh các tham số của mô hình, như kích thước lưới, ngưỡng tin cậy (confidence threshold), và ngưỡng IoU (Intersection over Union), để đảm bảo độ chính xác cao và giảm các dự đoán sai sót.

- Ứng dụng mô hình

- Sau khi đào tạo, triển khai mô hình YOLO vào ứng dụng xử lý ảnh để nhận diện và phân biệt các xe trong thời gian thực.

3.2.Thuật toán Sort (Simple Online and Realtime Tracking).

Thuật toán SORT là một thuật toán theo dõi đối tượng trong thời gian thực, phổ biến và hiệu quả. Dưới đây là mô tả về các bước chính của thuật toán SORT:

- KHỞI TẠO:
  - Khởi tạo danh sách trống để lưu trữ các đối tượng được theo dõi (tracks).

- NHẬN ĐIỂM DỮ LIỆU:
  - Nhận điểm dữ liệu mới từ hệ thống nhận diện vật thể, trong đó mỗi điểm dữ liệu thường chứa thông tin về vị trí (ví dụ: bounding box), đặc điểm của đối tượng và xác suất nhận diện.

- DỰ ĐOÁN VÀ CẬP NHẬT:
  - Dựa trên vị trí hiện tại của các tracks, thực hiện dự đoán vị trí tiếp theo của mỗi track sử dụng một mô hình dự đoán chuyển động (ví dụ: Kalman

Filter).

- TÍNH TOÁN ĐỘ TƯƠNG QUAN:
  - Tính toán độ tương quan giữa các track dự đoán và điểm dữ liệu mới. Điều này thường bao gồm việc sử dụng hàm chi phí (cost function) để đo lường sự tương quan giữa dự đoán và dữ liệu.

- GÁN NHÃN VÀ TÍNH TOÁN XÁC SUẤT:
  - Sử dụng thuật toán Hungarian (còn được gọi là Munkres) để gán nhãn các track với điểm dữ liệu mới dựa trên chi phí tương quan tính được. Tính toán xác suất để đánh giá sự đáng tin cậy của các gán nhãn.

- CẬP NHẬT TRACKS:
  - Dựa trên gán nhãn từ bước trước, cập nhật thông tin của các tracks, bao gồm vị trí, tốc độ, và các đặc điểm khác. Các tracks không có gán nhãn mới có thể được coi là đối tượng mới.

- QUẢN LÝ TRACKS:

- Loại bỏ các tracks không còn cần theo dõi hoặc không đáng tin cậy dựa trên một số tiêu chí như thời gian không nhận diện, xác suất thấp, hoặc vị trí nằm ngoài khu vực quan sát.

- TRẢ VỀ KẾT QUẢ:

- Trả về danh sách các tracks sau khi đã cập nhật, có thể được sử dụng để hiển thị hoặc phân tích tiếp theo.

Phần 4: Chương Trình.

- Khai báo thư viện:

```
import cv2
import math
import pandas as pd
import tkinter as tk
import cvzone
from datetime import datetime
from openpyxl import Workbook
from openpyxl.styles import Font
from PIL import Image, ImageTk
from ultralytics import YOLO
from tracker import*
```

- Model và video

```
# model và video
model=YOLO('best.pt')
cap=cv2.VideoCapture(r"C:\Users\Acer\Downloads\n1.mp4")
```

- Lưu thời gian bắt đầu chạy program

```
previous_time = datetime.now().date()
```

- Tạo file excel

```
workbook = Workbook()
sheet = workbook.active
sheet.append(["Thoi gian", "Xe dap", "Xe may", "O to", "Tổng"])
for cell in sheet["1:1"]:
    cell.font = Font(bold=True)
sheet.cell(row=2, column=1, value=previous_time)
sheet.cell(row=2, column=2, value=0)
sheet.cell(row=2, column=3, value=0)
sheet.cell(row=2, column=4, value=0)
sheet.cell(row=2, column=5, value=0)
```

- Phân loại tên xe

```
class_list = ["bicycle", "motorcycle", "car"]
```

- Tạo giao diện

```
window = tk.Tk()
window.title("Show")
window.geometry("1100x1080")
large_font = ('Times New Roman', 16)
label_camera1 = tk.Label(window)
label_camera1.place(x = 10, y = 10)
label1 = tk.Label(window, font = large_font)
label1.place(x=10, y=560)
label2 = tk.Label(window, text = "Time", font = large_font)
label2.place(x = 10, y = 520)
label3 = tk.Label(window, text = "Bicycle", font = large_font)
label3.place(x = 140, y = 520)
label4 = tk.Label(window, text = "Motorcycle", font = large_font)
label4.place(x = 220, y = 520)
label5 = tk.Label(window, text = "Car", font = large_font)
label5.place(x = 330, y = 520)
label6 = tk.Label(window, text = "Total", font = large_font)
label6.place(x = 370, y = 520)
label7 = tk.Label(window, font = large_font)
label7.place(x=145, y=560)
label8 = tk.Label(window, font = large_font)
label8.place(x=225, y=560)
label9 = tk.Label(window, font = large_font)
label9.place(x=335, y=560)
label10 = tk.Label(window, font = large_font)
label10.place(x=375, y=560)
```

- Các biến

```
count=0
cy1=424
offset=15
current_row = 2
counter1=[]
counter2=[]
counter3=[]
```

- **Trong vòng while**
- Sang ngày sẽ cập nhật lại dữ liệu

```
current_time = datetime.now().date()

label1.config(text=current_time)
if current_time > previous_time:
    current_row += 1
```

- Tạo dữ liệu cho phương tiện

```
a=results[0].boxes.data
px=pd.DataFrame(a).astype("float")
```

- Tạo các mảng rỗng

```
list1=[]
motorcycle=[]
list2=[]
car=[]
list3=[]
bicycle=[]
```

- Line đếm xe

```
cv2.line(frame,(100,cy1),(1100,cy1),(0,0,255),2)
```

- Gán nhãn cho phương tiện

```
for r in results:
    boxes = r.boxes
    for box in boxes:
        x1, y1, x2, y2 = box.xyxy[0]
        x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2)
        w, h = x2 - x1, y2 - y1
        cx, cy = x1+w//2, y1+h//2
        cv2.circle(frame,(cx,cy),5,(255,0,255),cv2.FILLED)
        conf = math.ceil((box.conf[0]*100)) / 100
        cls = int(box.cls[0])
        currentclass = class_list[cls]
        if currentclass in ["bicycle", "motorcycle", "car"] and conf > 0.5:
            cvzone.cornerRect(frame, (x1, y1, w, h), l=15, rt=2, colorR=(255,0,0))
            cvzone.putTextRect(frame,f'{class_list[cls]} {conf}', (max(0, x1), max(35, y1)), scale=0.6, thickness = 1, offset=3)
```

- Lưu các thông số trong hàng của các index vào danh sách của từng phương tiện tương ứng

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

    x1=int(row[0])
    y1=int(row[1])
    x2=int(row[2])
    y2=int(row[3])
    d=int(row[5])
    c=class_list[d]
    if 'motorcycle' in c:
        list1.append([x1,y1,x2,y2])
        motorcycle.append(c)
    if 'car' in c:
        list2.append([x1,y1,x2,y2])
        car.append(c)
    if 'bicycle' in c:
        list3.append([x1,y1,x2,y2])
        bicycle.append(c)
```

- Theo dõi các đối tượng

```
bbox1_idx=tracker1.update(list1)
bbox2_idx=tracker2.update(list2)
bbox3_idx=tracker3.update(list3)
```

- Đếm xe máy

```
for bbox1 in bbox1_idx:
    for i in motorcycle:
        x5,y5,x6,y6,id2=bbox1
        cxm=int(x5+x6)//2
        cym=int(y5+y6)//2
        if cym<(cy1+offset) and cym>(cy1-offset):
            cv2.circle(frame,(cxm,cym),4,(0,255,0),-1)

            if counter1.count(id2)==0:
                counter1.append(id2)
                cv2.line(frame,(100,cy1),(1100,cy1),(0,255,0),2)
                formatted_time = current_time.strftime("%Y-%m-%d")
                sheet.cell(row=current_row, column=1, value=formatted_time)
                sheet.cell(row=current_row, column=4, value=len(counter1))
```

- Đếm ô tô

```
for bbox2 in bbox2_idx:
    for i in car:
        x5,y5,x6,y6,id2=bbox2
        cxm=int(x5+x6)//2
        cym=int(y5+y6)//2
        if cym<(cy1+offset) and cym>(cy1-offset):
            cv2.circle(frame,(cxm,cym),4,(0,255,0),-1)

            if counter2.count(id2)==0:
                counter2.append(id2)
                cv2.line(frame,(100,cy1),(1100,cy1),(0,255,0),2)
                formatted_time = current_time.strftime("%Y-%m-%d")
                sheet.cell(row=current_row, column=1, value=formatted_time)
                sheet.cell(row=current_row, column=4, value=len(counter2))
```

- Đếm xe đạp

```
for bbox3 in bbox3_idx:
    for i in bicycle:
        x5,y5,x6,y6,id2=bbox3
        cxm=int(x5+x6)//2
        cym=int(y5+y6)//2
        if cym<(cy1+offset) and cym>(cy1-offset):
            cv2.line(frame,(2,cy1),(794,cy1),(0,255,0),2)

            cvzone.putTextRect(frame,f'{id2}',(x5,y5),1,1)
            if counter3.count(id2)==0:
             counter3.append(id2)
             cv2.line(frame,(100,cy1),(1100,cy1),(0,255,0),2)
             formatted_time = current_time.strftime("%Y-%m-%d")
             sheet.cell(row=current_row, column=1, value=formatted_time)
             sheet.cell(row=current_row, column=4, value=len(counter3))
```

- Cập nhật dữ liệu

```
motorcyclec=(len(counter1))
carr=(len(counter2))
bicyclee=(len(counter3))
s = (len(counter1))+(len(counter2))+(len(counter3))
label7.config(text=bicyclee)
label8.config(text=motorcyclec)
label9.config(text=carr)
label10.config(text=s)
photo = ImageTk.PhotoImage(Image.fromarray(frame))
label.config(image=photo)
label.photo = photo
```

- Lưu dữ liệu lên excel

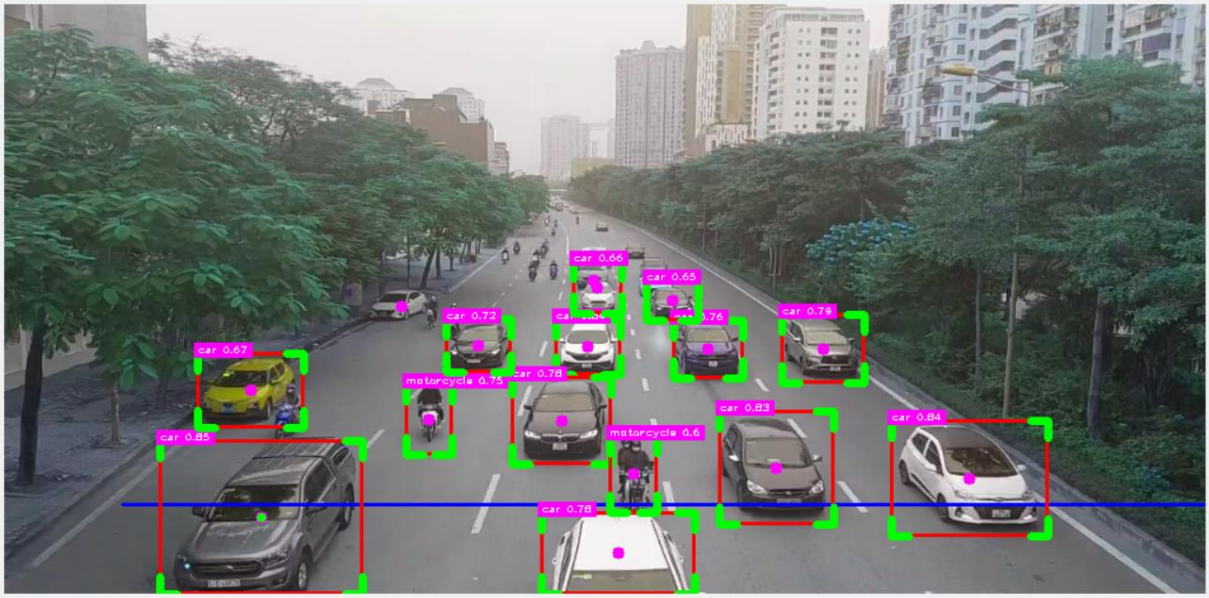
```
sheet.cell(row=current_row, column=5, value=s)
workbook.save("xlad9.xlsx")
```

Phần 5:Kết Luận.

5.1. Kết quả đạt được.

- Chương trình đã đạt được các kết quả sau:
  - Nhận diện 3 phương tiện thành công từ video.
  - Đếm số xe chính xác 99%.

  Kết luận: đã đáp ứng được yêu cầu bài ra một cách hoàn chỉnh. Sau đây là vài hình ảnh cho minh họa cho chương trình



Time	Bicycle	Motorcycle	Car	Total
2023-12-14	0	4	8	12

- Dữ liệu:được lưu thành 1 file excel

Thời gian	Xe đạp	Xe máy	Ô tô	Tổng
12/12/2023	4	10	26	40
13/12/2023	0	5	13	18
14/12/2023	2	8	19	29

5.2. Vấn đề cần giải quyết.

Chương trình trên còn tồn tại một số vấn đề cần được cải thiện và giải quyết:

- Thuật toán này cần phải cài đặt nhiều ứng dụng (ví dụ CUDA cho phép chạy GPU nếu không chương trình chạy rất chậm).
- Khi sử dụng YOLO với độ phân giải cao, mô hình có thể trở nên khá phức tạp, gây tốn nhiều tài nguyên tính toán.

Sour-code:
https://drive.google.com/drive/folders/1i-aASArrcMM-gGp5zX2SuioZtAFSDRMc?usp=sharing