

# بسم الله الرحمن الرحيم

موضوع پروژه: شمارش تعداد افراد روی دوچرخه

ارائه دهندگان : سوگند محمد خاتمی ، مریم خالقی

مراحل انجام پروژه :

- نصب **vscode** : ویژوال استودیو کد یک ویرایشگر کد یا به اصطلاح کدایتور قدرتمند و بسیار محبوب از مایکروسافت است که می توان آن را در موارد گسترده ای از کارهای مربوط به برنامه نویسی خود به کار بگیریم. صرف نظر از اینکه فردی مبتدی هستیم یا یک برنامه نویس ماهر و باتجربه VS Code می تواند ما را در امر کدنویسی ، آزمایش و دیباگ مؤثر کدهایمان یاری کند .
- نصب کتابخانه **opencv** : با دستور `pip install opencv-python` این کتابخانه نصب میکنیم تا اینکه بتوانیم عکس ها رو بخوانیم و باز کنیم .
- نصب کتابخانه **pytorch** : با دستور `pip install torch torchvision torchaudio` این کتابخانه یه کتابخانه پیش فرض برای کتابخانه YOLO است. اجرای مدل هوش مصنوعی YOLO
- نصب کتابخانه **YOLOv8**: با دستور `pip install ultralytics` مدلی برای تشخیص اشیا که برای تشخیص اسان افراد و دوچرخه و اشیا و...
- نصب کتابخانه **media pipe**: برای تشخیص وضعیت بدن در تصاویر
- در **vscode** برای نصب کتابخانه ها از منوی **terminal** گزینه **new terminal** را میزنیم پایین صفحه قسمتی باز میشه و ما در آنجا دستورات را وارد میکنیم تا کتابخانه های مورد نیازمان را نصب کنیم
- پس از نصب کتابخانه ها از منوی **File** گزینه **open folder** را میزنیم و یک فولدر برای کارمان انتخاب میکنیم و یک پروژه با نام **bike-count-image** باز میکنیم .
- در اینجا کدهای مورد نیاز پروژه را وارد میکنیم .

```
from ultralytics import YOLO
import cv2

# آماده YOLOv8 بارگذاری مدل
model = YOLO("yolov8n.pt")

# مسیر عکس
image_path = "test.jpg" # مطمئن شو که این عکس داخل پوشه پروژه هست
image = cv2.imread(image_path)

# اجرای مدل روی عکس
results = model(image)[0]

# جدا کردن اشیاء تشخیص داده شده
boxes = results.boxes

# لیست جدا برای افراد و دوچرخه ها
persons = []
bicycles = []

for box in boxes:
    cls = int(box.cls[0])
    if cls == 0: # کلاس ۰ = شخص
        persons.append(box)
    elif cls == 1: # کلاس ۱ = دوچرخه
        bicycles.append(box)

# شمارش افرادی که روی دوچرخه هستن
count = 0

for person in persons:
    px1, py1, px2, py2 = person.xyxy[0]
    for bike in bicycles:
        bx1, by1, bx2, by2 = bike.xyxy[0]

        # محاسبه مرکز بدن فرد
        px_center = (px1 + px2) / 2
        py_center = (py1 + py2) / 2

        # آگه مرکز بدن فرد داخل محدوده دوچرخه باشه
        if bx1 < px_center < bx2 and by1 < py_center < by2:
            count += 1
```

break

```
print(f"تعداد افرادی که روی دوچرخه هستند: {count}")  
# این کد برای تشخیص افراد سوار بر دوچرخه هستند
```

---

- توضیحات کد :

۱. وارد کردن کتابخانه‌ها:

```
from ultralytics import YOLO  
import cv2
```

ultralytics: کتابخانه‌ای برای استفاده از مدل‌های YOLOv۸.

cv2: کتابخانه OpenCV برای کار با تصاویر.

۲. بارگذاری مدل YOLO:

```
model = YOLO("yolov8.pt")
```

مدل YOLO نسخه‌ی سبک (nano) رو بارگذاری می‌کنه. این مدل می‌تونه اشیاء مختلف مثل "شخص"، "دوچرخه"، "ماشین" و ... رو تشخیص بده.

۳. مشخص کردن مسیر عکس:

```
image_path = "test.jpg"
```

آدرس فایل‌ی که می‌خواه روی اون پردازش انجام بدی. عکس باید در کنار همین فایل پایتون باشه.

۴. خواندن عکس:

```
image = cv2.imread(image_path)
```

عکس رو از مسیر مشخص شده می‌خونه و در متغیر image ذخیره می‌کنه.

۵. اجرای مدل روی عکس:

```
results = model(image)
```

مدل YOLO روی عکس اجرا می‌شه و اشیاء موجود در عکس رو تشخیص می‌ده.

۶. گرفتن جعبه‌های شناسایی‌شده:

```
]boxes classes = boxes.cls xyxy = boxes.xyxy • boxes = results[
```

`boxes`: جعبه‌هایی که دور اشیاء تشخیص داده شده کشیده شده.

`classes`: شماره کلاس هر جعبه (۰ برای شخص، ۱ برای دوچرخه و ...).

`xyxy`: مختصات گوشه‌های بالا-چپ (۱,۷۱X) و پایین-راست (۲,۷۲X) جعبه.

۷. جدا کردن افراد و دوچرخه‌ها:

```
persons = [] bicycles = [] for i in range(len(classes)): class_id = int(classes[i]) box = xyxy[i]
: bicycles.append(box) \: persons.append(box) elif class_id == ۰ if class_id ==
```

این حلقه، جعبه‌هایی که کلاس‌شون "شخص" یا "دوچرخه" هست رو در دو لیست جدا ذخیره می‌کنه.

۸. تعریف تابع برای بررسی اینکه فرد روی دوچرخه هست یا نه:

```
۵۰def is_on_bike(person, bike, threshold=
```

این تابع بررسی می‌کنه که آیا یک فرد خاص روی یک دوچرخه خاص قرار گرفته یا نه.

داخل تابع چند شرط بررسی می‌شن:

فاصله‌ی افقی مرکز شخص و دوچرخه کم باشه.

شخص پایین‌تر از قسمت بالای دوچرخه باشه.

جعبه‌های شخص و دوچرخه هم‌پوشانی داشته باشن.

ارتفاع شخص خیلی زیاد نباشه (یعنی نشسته باشه نه ایستاده).

۹. شمارش تعداد افرادی که فقط روی دوچرخه هستند:

```
for person in persons:
    for bike in bicycles:
        if is_on_bike(person, bike):
            count += 1
        count = break ۱
```

برای هر شخص بررسی می‌کنه که آیا روی یک دوچرخه نشسته یا نه. اگر بود، شمارنده رو افزایش می‌ده و می‌ره سراغ فرد بعدی.

۱۰. چاپ نتیجه:

```
print(f"تعداد افرادی که فقط روی دوچرخه هستند: {count}")
```

تعداد افرادی که روی دوچرخه نشسته‌اند رو نمایش می‌ده.

- 
- تست عکس : در اینجا عکس موردنظرمان را با نام test.jpg ذخیره میکنیم و داخل فایللی که ساختیم میزاریم تا در زمان اجرا طبق کد از مسیر مشخص شده عکس را چک کند.
  - اجرا: با دستور python bike-count-image.py پروژه‌ها اجرا میگیریم و نتیجه که چند نفر روی دوچرخه سوار هستند را به ما میدهد.
  -