پروژه تشخیص و شمارش کفش‌ها با استفاده از پردازش تصویر

استاد مجتبی فر

گرد آورندگان: ساریت دانیال مفرد – پریناز نظری

فهرست مطالب

1.معرفی کلی

2.پیش‌نیازها

3.نصب و راه‌اندازی

4.توضیحات خط به خط کد

5.نمونه خروجی

6.راهکارهای بهبود

7.گزینه جایگزین (YOLO)

8.نتیجه گیری

معرفی کلی

در این پروژه، از تکنیک‌های پردازش تصویر و بینایی ماشین برای تشخیص و شمارش کفش‌ها در یک تصویر استفاده شده است. این روش می‌تواند در صنایع مختلف مانند فروشگاه‌های آنلاین، انبارداری و سیستم‌های کنترل کیفیت کاربرد داشته باشد.

این برنامه یک اسکریپت پردازش تصویر است که از کتابخانه OpenCV برای تشخیص و شمارش کفش‌ها در یک تصویر استفاده می‌کند. روش کار مبتنی بر پردازش تصویر کلاسیک با مراحل زیر است:

1. خواندن تصویر ورودی

2. تبدیل به تصویر خاکستری

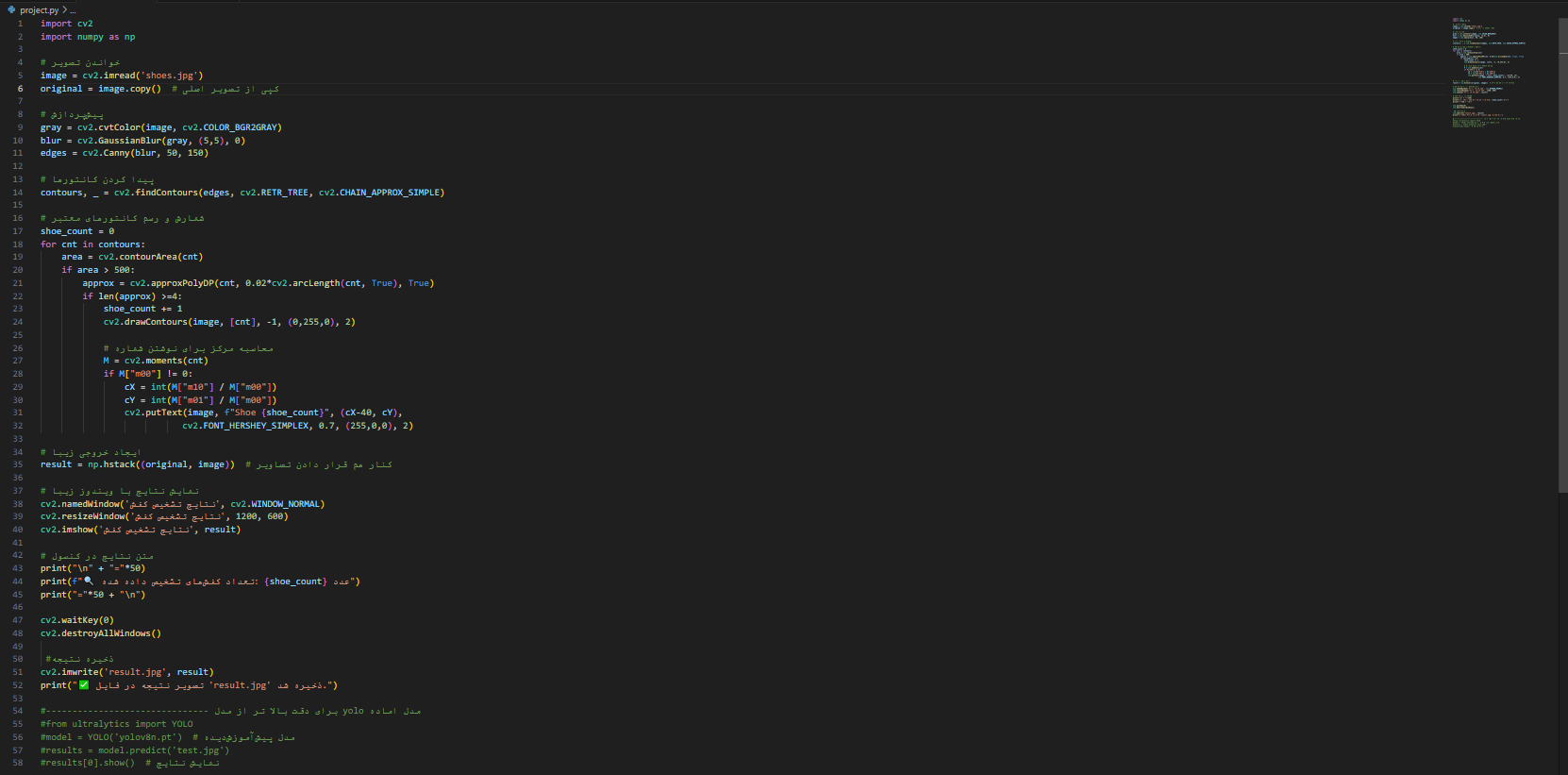
3. اعمال فیلتر گوسی برای کاهش نویز

4. تشخیص لبه‌ها با الگوریتم Canny

5. یافتن کانتورها (نقاط مرزی اشیا)

6. فیلتر کردن کانتورهای معتبر بر اساس مساحت و شکل

7. شمارش و علامت‌گذاری کفش‌های تشخیص داده شده



نگاه کلی به کد ها

پیش‌نیازها

برای اجرای این کد نیاز به نصب کتابخانه‌های زیر دارید:

pip install opencv-python numpy

برای گزینه YOLO که در بخش پایانی آمده:

pip install ultralytics

نصب و راه‌اندازی

1. کد را در یک فایل با نام shoe\_detection.py ذخیره کنید.

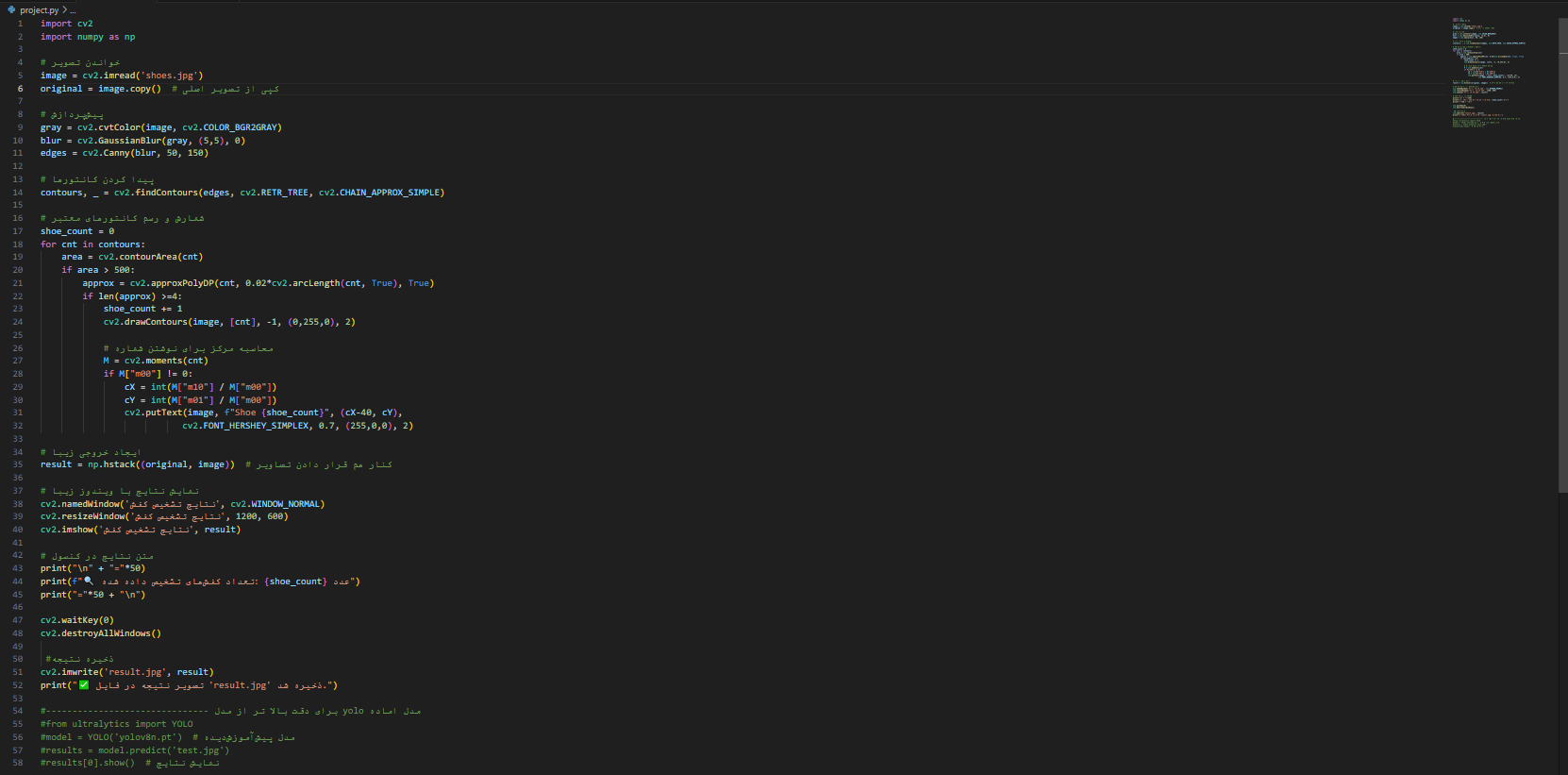
2. تصویری از کفش‌ها با نام shoes.jpg در همان مسیر قرار دهید.

3. صفحه کد خود را بسازید کنید:

python shoe\_detection.py

توضیحات خط به خط کد

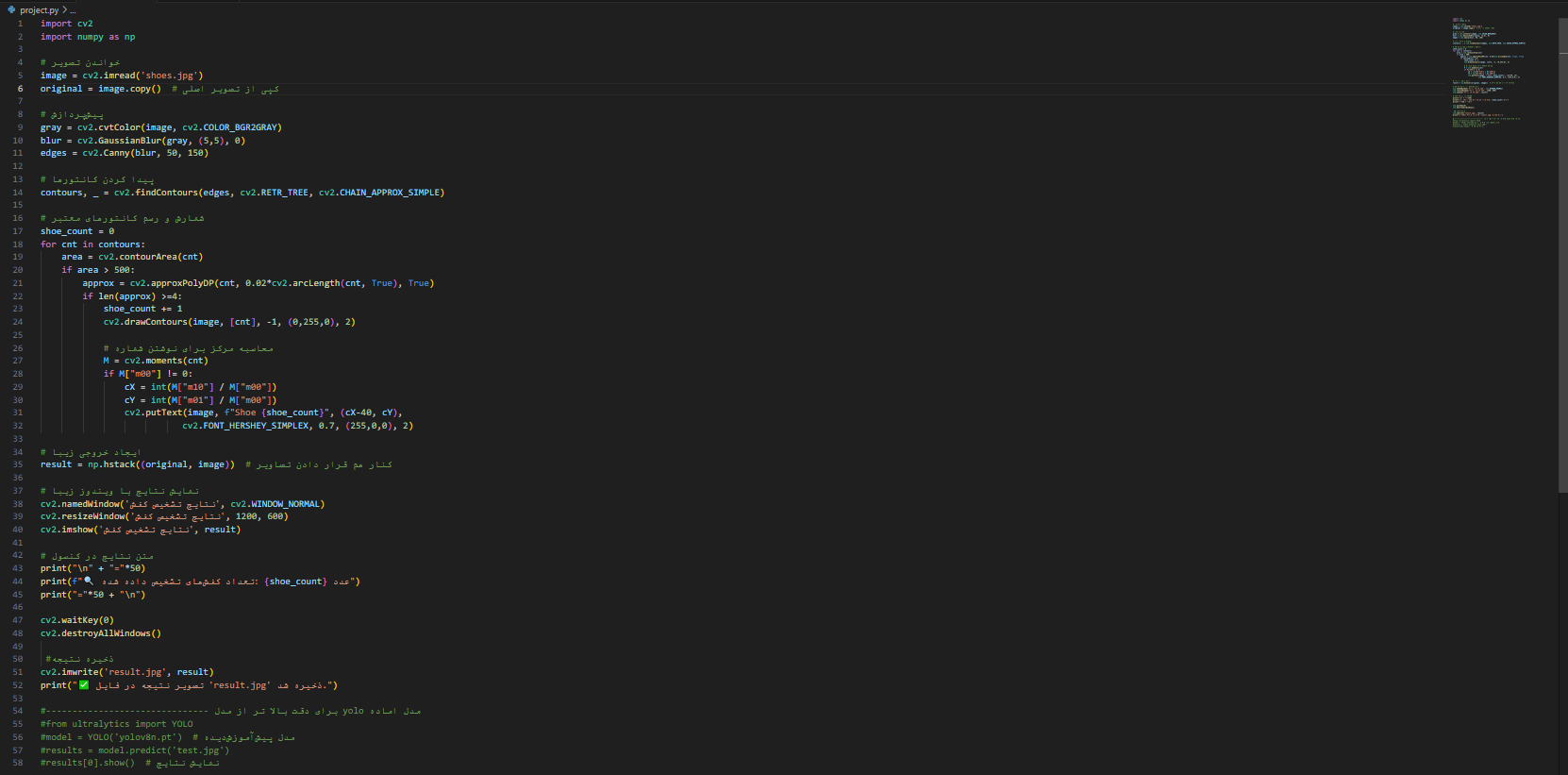
بخش 1: وارد کردن کتابخانه‌ها



Cv2 -کتابخانه OpenCV برای پردازش تصویر

Numpy -برای محاسبات عددی و کار با آرایه‌ها

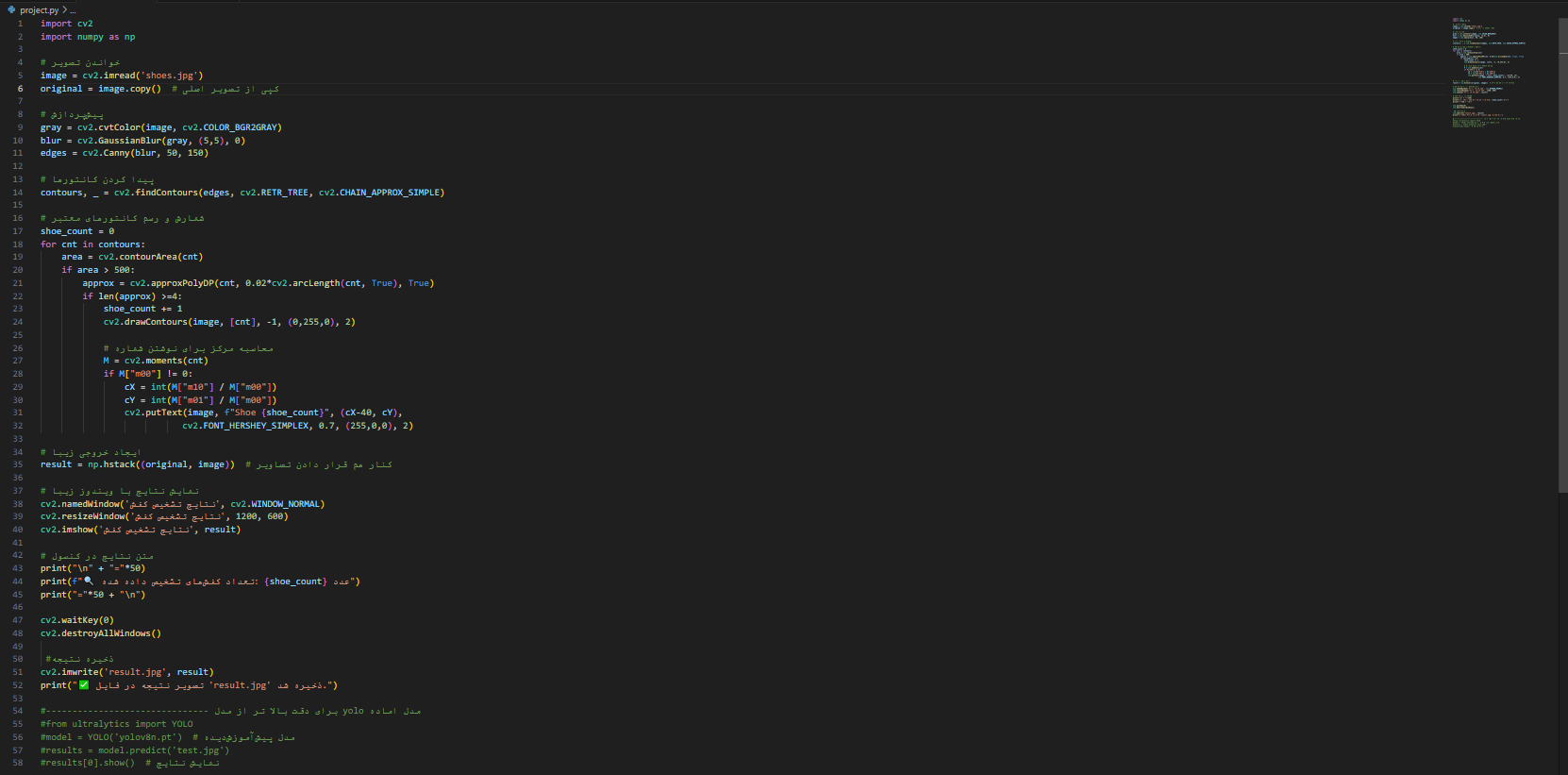
بخش 2: خواندن تصویر



- خواندن تصویر از فایل shoes.jpg

- ایجاد یک کپی از تصویر اصلی برای نمایش مقایسه‌ای در خروجی

بخش 3: پیش‌پردازش تصویر

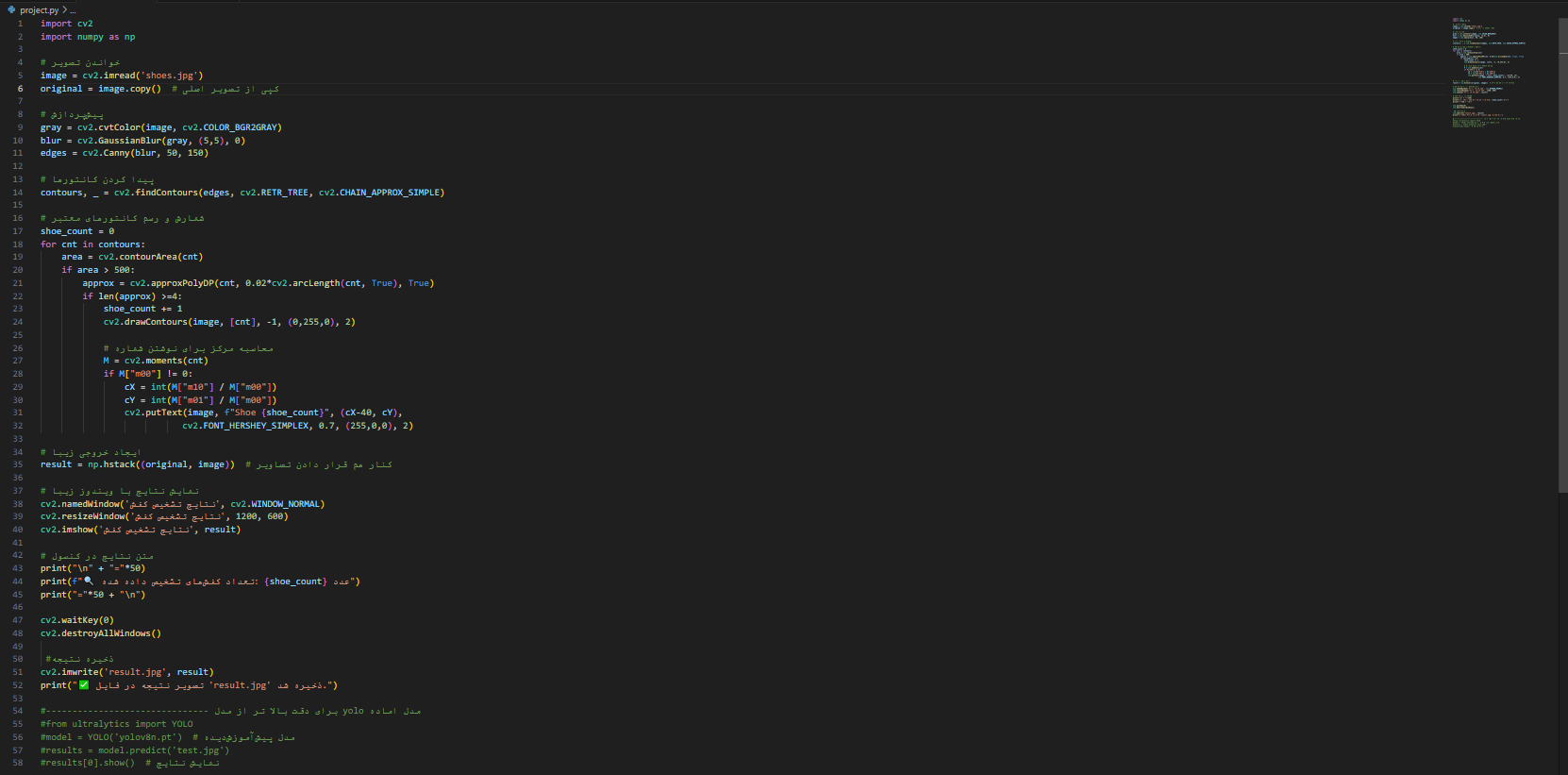


-تبدیل تصویر رنگی به خاکستری

-اعمال فیلتر گوسی با هسته 5x5 برای کاهش نویز

-تشخیص لبه‌ها با الگوریتم Canny با آستانه‌های 50 و 150

بخش 4: یافتن کانتورها

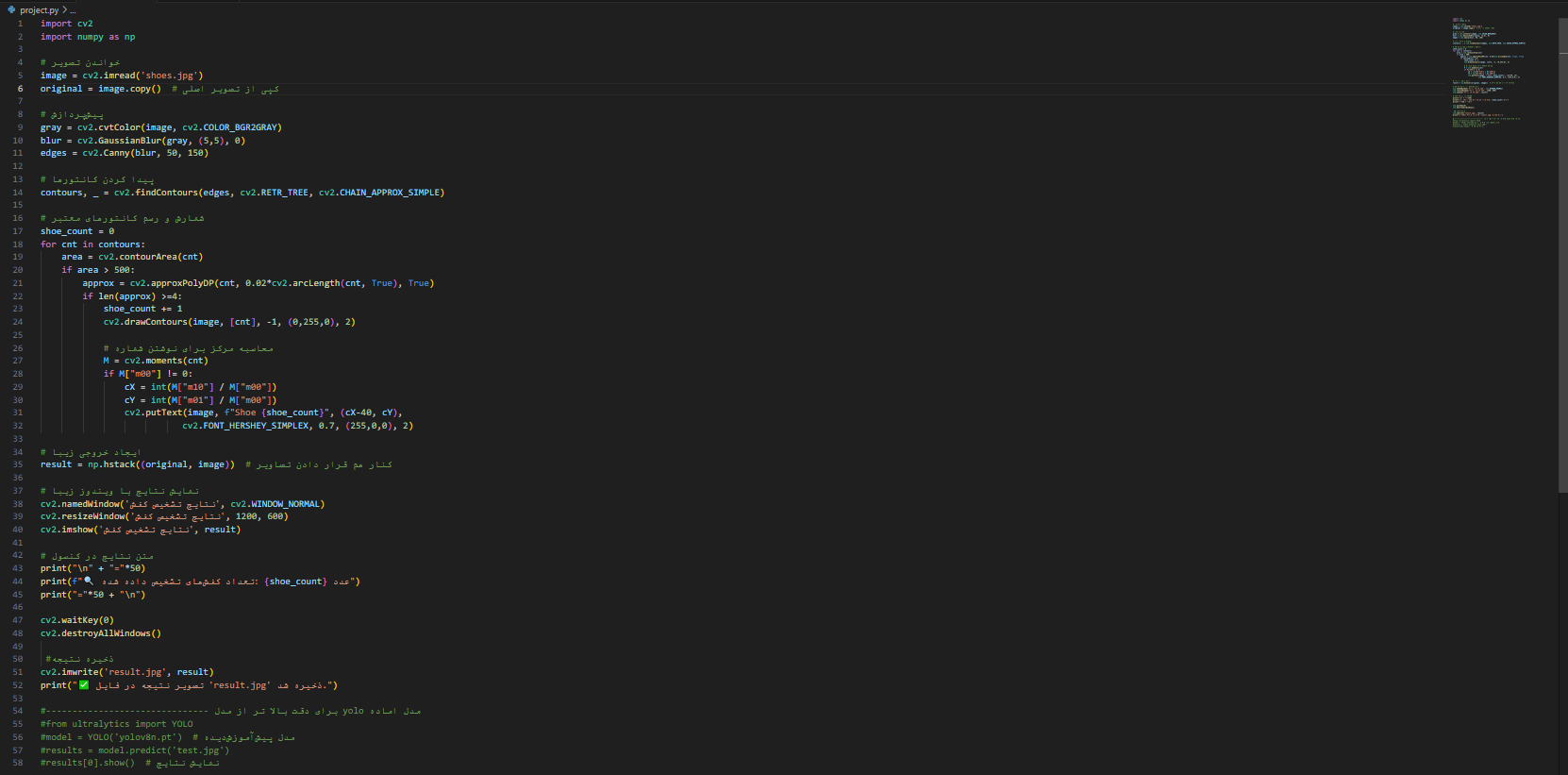


- یافتن تمام کانتورها در تصویر لبه‌ها

RETR\_TREE - بازیابی سلسله مراتبی کانتورها

CHAIN\_APPROX\_SIMPLE - فشرده‌سازی کانتورها با حذف نقاط اضافی

بخش 5: پردازش کانتورها و شمارش کفش‌ها



- شمارنده کفش‌ها را از صفر شروع می‌کنیم

- برای هر کانتور:

- محاسبه مساحت کانتور

- اگر مساحت بیشتر از 500 پیکسل بود:

- تقریب کانتور به یک چندضلعی ساده‌تر

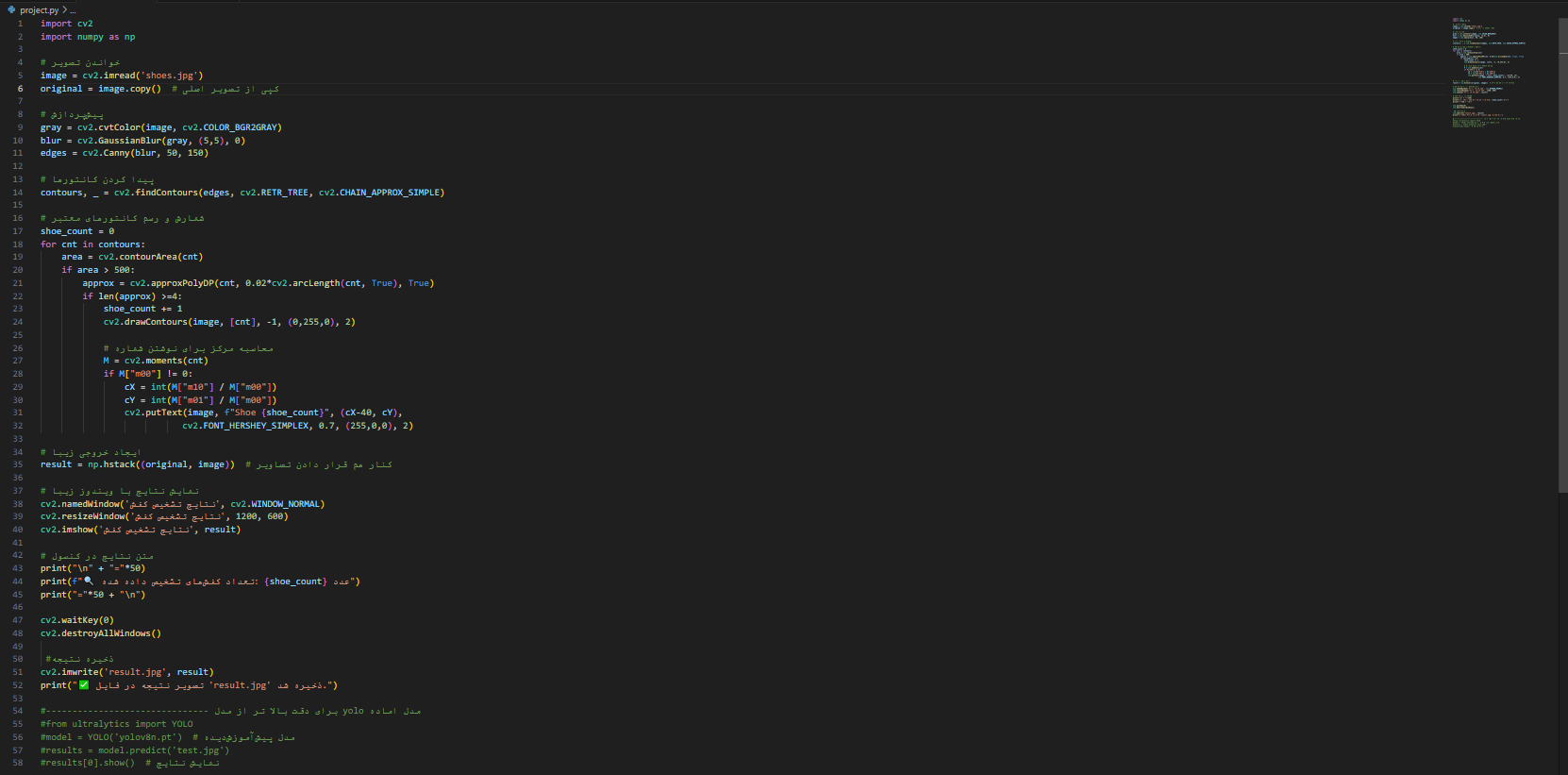
- اگر شکل حداقل 4 ضلع داشت (شرایط کلی برای اشیاء چهارگوش مانند کفش):

- افزایش شمارنده

- رسم کانتور روی تصویر با رنگ سبز

- محاسبه مرکز کانتور و نوشتن شماره کفش

بخش 6: نمایش نتایج

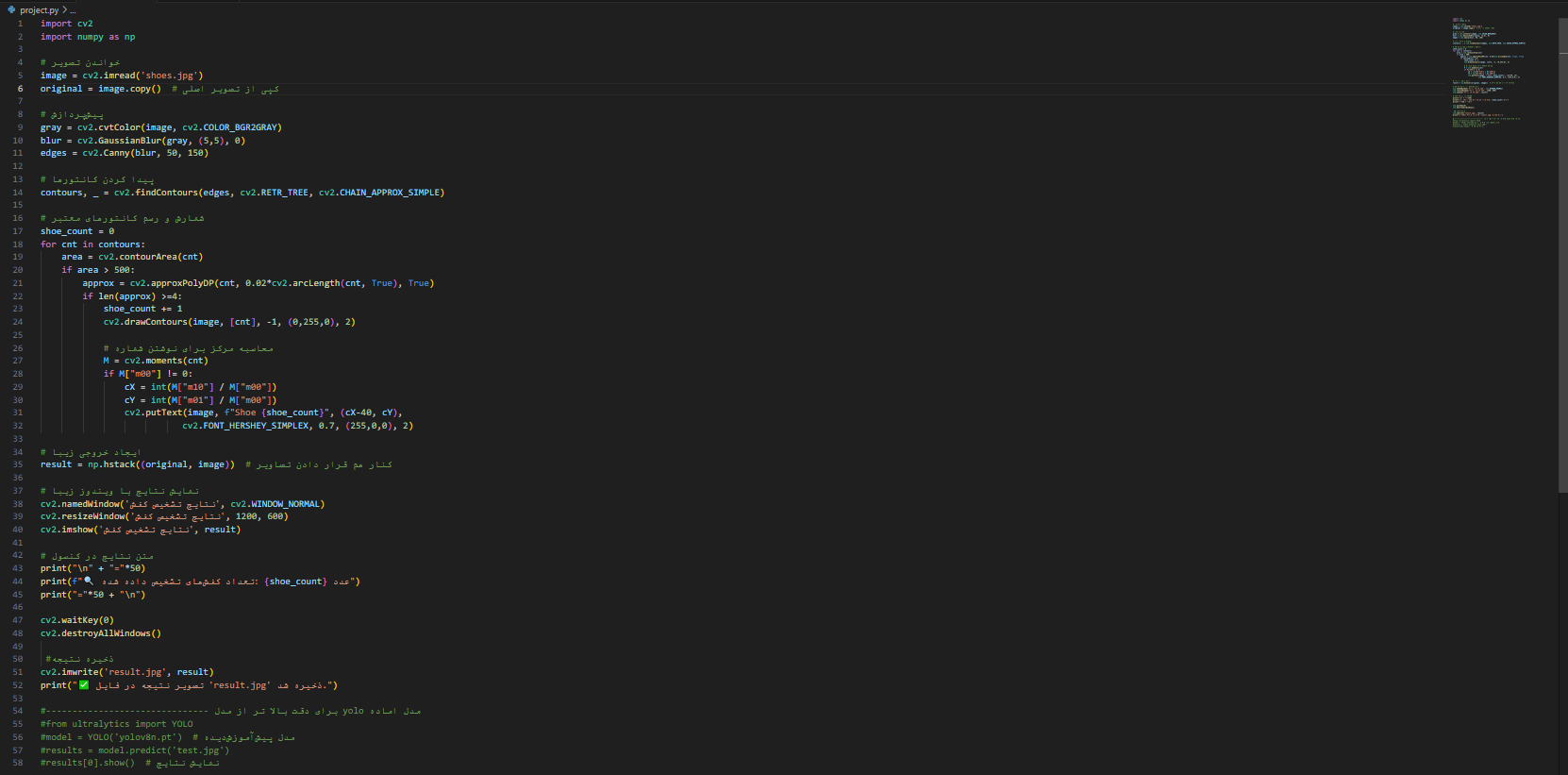


- کنار هم قرار دادن تصویر اصلی و نتیجه پردازش

- ایجاد پنجره قابل تنظیم سایز برای نمایش نتایج

- نمایش تصویر ترکیبی

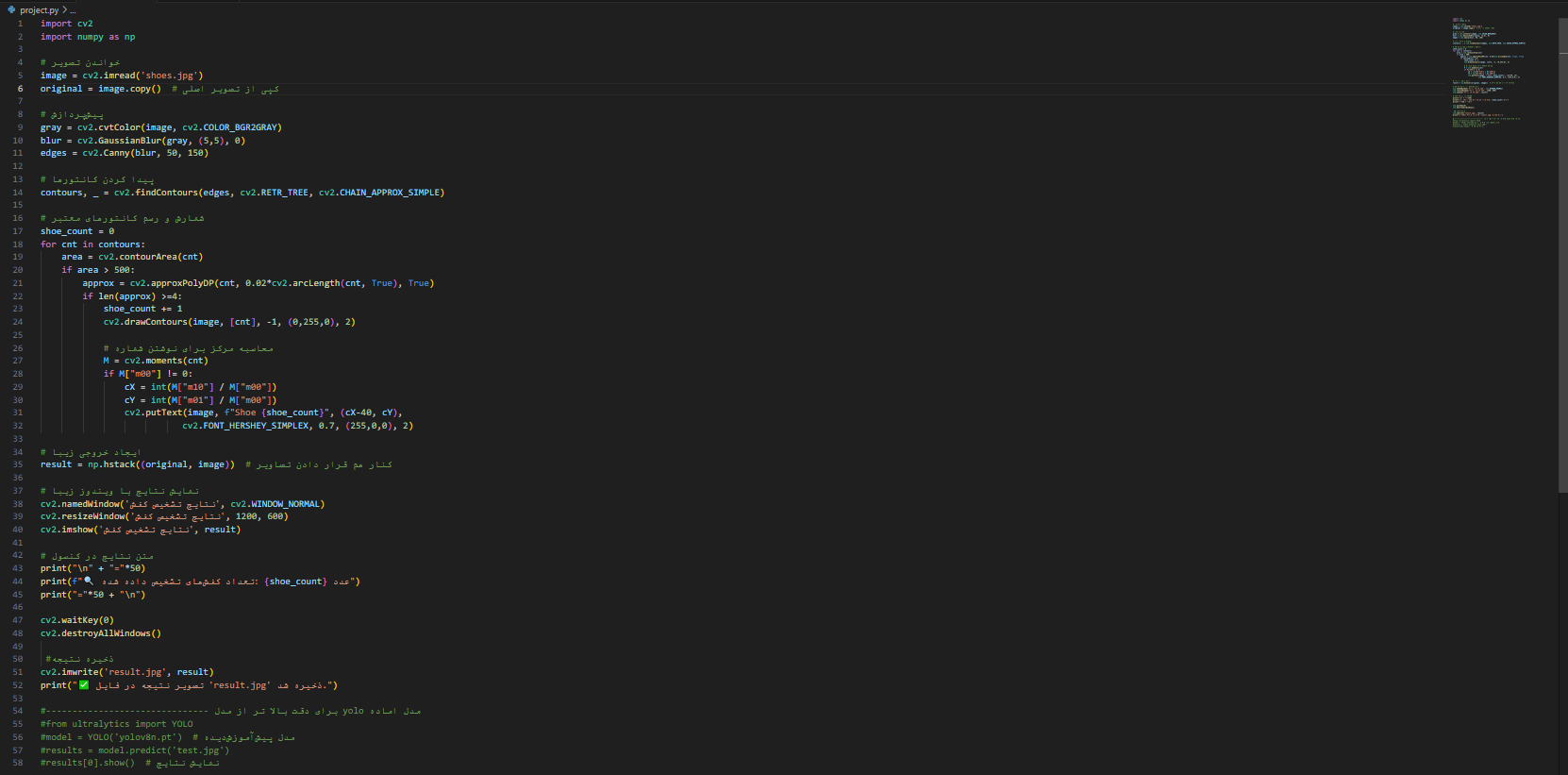
بخش 7: خروجی متنی و ذخیره نتایج



- نمایش تعداد کفش‌های تشخیص داده شده در کنسول

- ذخیره تصویر نتیجه در فایل result.jpg

بخش 8: گزینه پیشرفته (YOLO)



- این بخش به عنوان یک گزینه پیشرفته با دقت بالاتر (اما نیاز به منابع بیشتر) ارائه شده

- با حذف کامنت‌ها می‌توان از مدل YOLOv8 برای تشخیص اشیا استفاده کرد

نمونه خروجی

خروجی برنامه شامل:

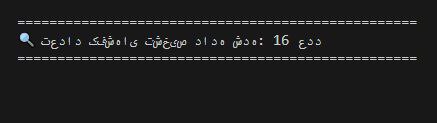
1. یک تصویر با دو بخش:

- سمت چپ: تصویر اصلی

- سمت راست: تصویر با کانتورهای تشخیص داده شده و شماره‌گذاری

2. خروجی کنسول که تعداد کفش‌ها را گزارش می‌دهد

مثال خروجی کنسول:

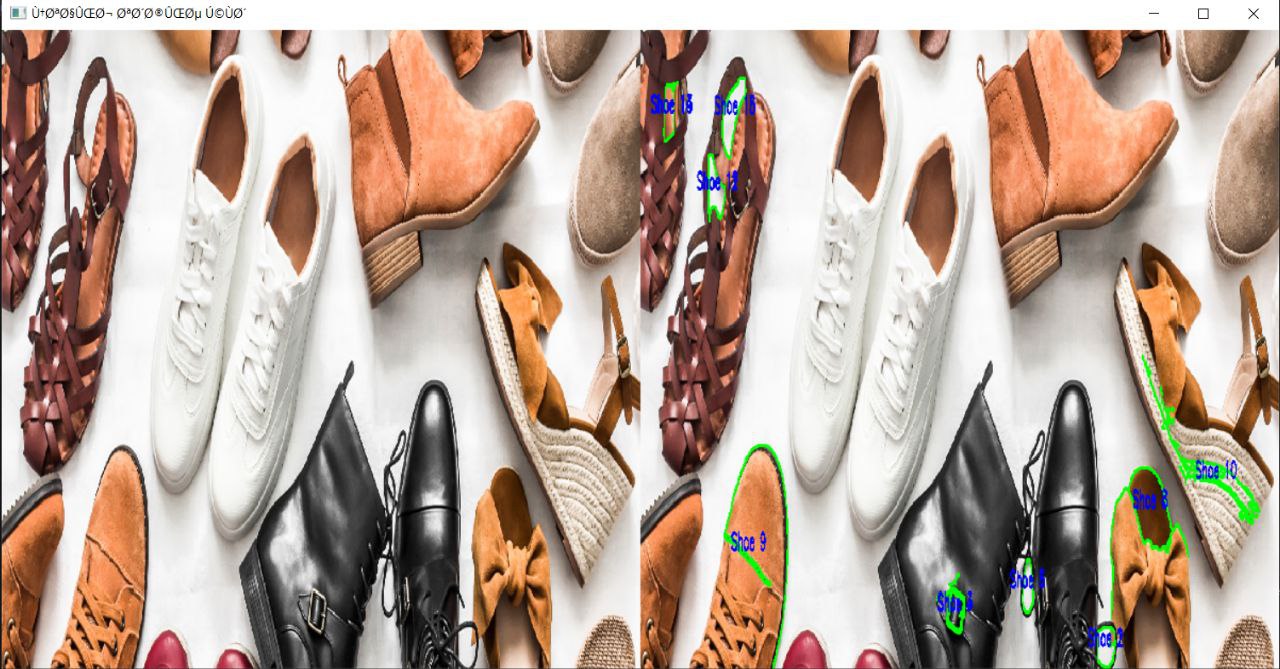


==================================================

🔍 تعداد کفش‌های تشخیص داده شده : 16 عدد

==================================================

تصویر:



راهکارهای بهبود

1.تنظیم پارامترها:

- آستانه‌های Canny (50, 150) را می‌توان برای تصاویر مختلف تنظیم کرد

- حداقل مساحت (500) بسته به اندازه کفش‌ها در تصویر قابل تغییر است

2. پیش‌پردازش بهتر:

مثال: افزایش کنتراست قبل از تشخیص لبه

alpha = 1.5 کنترل کنتراست

beta = 0 کنترل روشنایی

adjusted = cv2.convertScaleAbs(gray, alpha=alpha, beta=beta)

3. فیلتر کردن رنگ:

اگر کفش‌ها رنگ خاصی دارند

hsv = cv2.cvtColor(image, cv2.COLOR\_BGR2HSV)

lower\_color = np.array([0,50,50])

upper\_color = np.array([20,255,255])

mask = cv2.inRange(hsv, lower\_color, upper\_color)

4. بهبود شرایط تشخیص شکل:

شرایط دقیق‌تر برای شکل کفش

if len(approx) >=4 and len(approx) <=6:

گزینه جایگزین (YOLO)

برای دقت بالاتر می‌توان از مدل‌های یادگیری عمیق مانند YOLO استفاده کرد:

from ultralytics import YOLO

بارگذاری مدل پیش‌آموزش‌دیده

model = YOLO('yolov8n.pt')

انجام پیش‌بینی

results = model.predict('shoes.jpg')

نمایش نتایج

results[0].show()

مزایای YOLO

- دقت بسیار بالاتر

- توانایی تشخیص انواع مختلف اشیا

- تشخیص حتی وقتی اشیا همپوشانی دارند

معایب YOLO

- نیاز به منابع محاسباتی بیشتر

- زمان پردازش طولانی‌تر

- نیاز به اینترنت برای دانلود مدل در اولین اجر

نتیجه‌گیری

این پروژه نشان‌دهنده یک روش ساده ولی مؤثر برای تشخیص و شمارش کفش‌ها با استفاده از پردازش تصویر است. در حالی که روش مبتنی بر کانتورها برای تصاویر ساده کارآمد است، استفاده از یادگیری ماشین می‌تواند دقت را به‌طور چشمگیری افزایش دهد.