

— بسم الله الرحمن الرحيم —

## تمرین تحویلی شماره 2 - Multimedia

زهره لطیفی 9923069

i. در تصویر **AerialView.jpeg** تصویر چندین مرزعه برنج را مشاهده می‌کنید که مرزبندی مشخصی با یکدیگر دارند. (تصویر از **Gettyimages**) با توجه به مواردی که در مبحث **contour** ها مطالعه کردید، این مرزبندی ها را مشخص کنید و مزارع برنج را از یکدیگر جدا کنید. همچنین در مرحله بعد، مزارع را به ترتیب مساحت گزارش کنید. (**sort by area**) برای راحتی بیشتر **Watermark** تصویر تا حد امکان حذف شده است.

برای حل این سوال ابتدا تابع **imshow** را تعریف کردیم که برای **Load** کردن تصویر ورودی، نمایش هر تصویر در سبزه مورد نظر و قرار دادن عنوان مناسب استفاده می‌شود.

سپس تصویر مزارع را **Load** کردیم.

درحین انجام این سوال به این مشکل خوردیم که مرزها به وضوح دیتکت نمیشد؛ به همین دلیل یک کرنل تعریف کرده و به کمک آن تصویر را **Sharp** تر کردیم. همینطور در راستای همان منظور، **Contrast** و **Brightness** تصویر را هم تغییر دادیم.

در ادامه برای پیدا کردن کانتورها در تصویر، لازم بود از فیلتر **Canny** استفاده کنیم. پس هم تصویر را **GrayScale** کرده و هم باتوجه به تجربه سوال چهارم تمرین سری اول، از فیلتر **Gaussian** پیش از استفاده از **Canny** بهره بردیم تا لبه‌ها بهتر یافت شوند.

با جلو رفتن سوال، برای گرفتن نتایج بهتر مجبور شدیم از **Morphology** هم کمک گرفته و توابع **Erode** و **Dilate** هم مورد استفاده قرار دهیم. برای هر کدام از این‌ها، کرنلی از نوع **Ones** با آزمون و خطا ساختیم.

در نهایت **Canny** را اعمال کرده و نتیجه را نمایش دادیم. (در فایل **Q1\_HW2\_9923069** تمرین آپلود شده می‌توانید نتایج هربخش را مشاهده کنید.)

کانتورها را با تابع **cv2.findContours** یافتیم. به عنوان ورودی این تابع از تقریب **cv2.CHAIN\_APPROX\_NONE** استفاده کردیم که با توجه به اینکه کانتورها خطوط مرزی یک شکل با **intensity** یکسان هستند، مختصات  $(x,y)$  تمام نقاط مرزی یک شکل را ذخیره می‌کند.

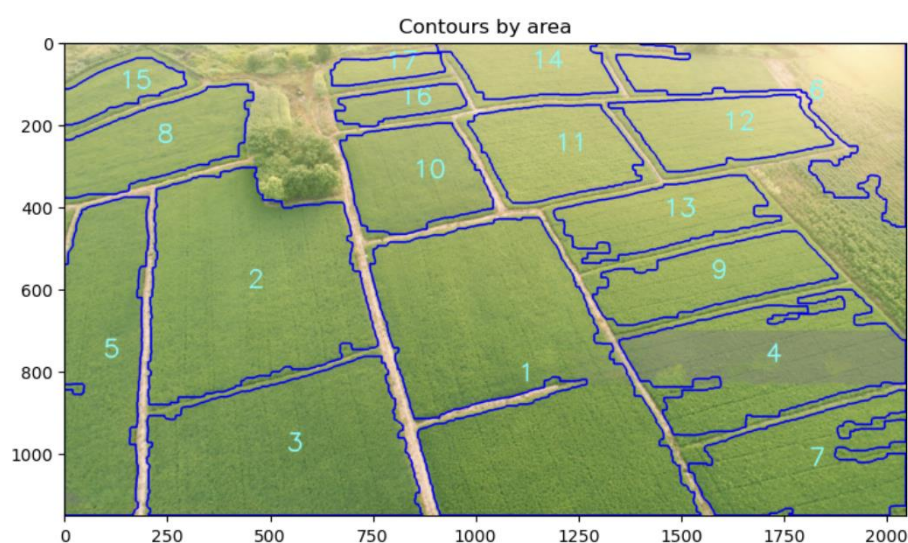
سپس با تابع `cv2.drawContours` محل کانتورهای یافت شده را بر روی شکل پررنگ کردیم.

در اینجا برخی کانتورها `detect` شده بود که مطلوب ما نبود و مرز مزرعه‌ای را مشخص نمی‌کرد. خوشبختانه ابعاد آن‌ها از مزارع کوچکتر بود و با اعمال یک شرط، کانتورهایی که مساحت کمتر از 12000 داشتند، حذف شدند.

باردیگر شکل با مشخص شدن کانتورها نمایش داده شده و تعداد کانتورهای یافت شده (17) هم گزارش شده است.

در این مرحله تابعی تعریف کردیم که در آن مساحت هر کانتور با دستور `cv2.contourArea` محاسبه شده و در آرایه‌ای ذخیره می‌شوند و سپس به ترتیب آن‌ها را `Sort` کردیم و شماره‌ای به هر مساحت اختصاص یافت.

در نهایت با دستور `moment`، مرکز هر مزرعه را یافته و شماره آن را روی آن نوشتیم.



ii. در تصویر **airport.jpg** تصویر هوایی از یک فرودگاه را مشاهده می‌کنید. با استفاده از **Template Matching** هواپیما های موجود در تصویر را پیدا کنید.

تصویر فرودگاه را به همان ترتیب سوال قبل، **Load** کردیم. سپس آن را **GrayScale** کرده و تصویر یکی از هواپیما ها را به عنوان **template** قرار دادیم. (این تصویر را هم **gary scale** کردیم).

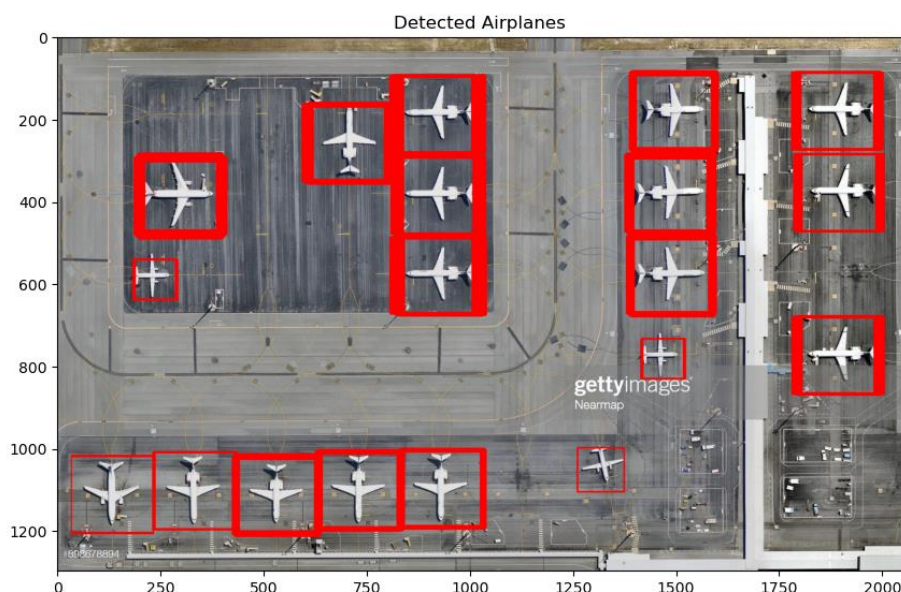


با کمک دستور **cv2.matchTemplate** و **threshold** برابر با **0.35**، محل هواپیماها را روی تصویر اصلی یافتیم و با پوینتر روی این آرایه ایجاد شده حرکت کرده و مستطیل هایی را رسم کردیم که محل هواپیما ها را در ابعاد تصویر **template** مشخص می‌کنند. در این مرحله سه هواپیمای کوچکتر پیدا نشدند. پس با **cv2.resize** ابعاد **template** را **0.45** برابر کرده و دیگر بار همان کار را تکرار کردیم.

در این مرحله باز هواپیمایی که مقداری زاویه داشت یافت نشد که برای حل این مشکل، این بار **template** کوچک شده را به اندازه **290** درجه چرخانیدیم و همان مراحل را تکرار کردیم. مشکل حل شد و تمام هواپیماها **Detect** شدند.

توجه داریم که اعداد اضافه شده به **w** و **h** در رسم مستطیل ها، تنها برای فیت شدن آنها بر هواپیماهاست زیرا ابعاد اندک متفاوتی داشتند.

با مراجعه به فایل **Q2\_HW2\_9923069** از تمرین آپلود شده، نتایج هرمرحله قابل مشاهده است.



iii. در اسلاید آخر مبحث **Template Matching** تصویر یک قفسه فروشگاه قرار دارد. 2 محصول از بین آن ها را انتخاب کرده و تعداد آن ها را به صورت جداگانه گزارش کنید.

برای حل این سوال ابتدا تصویر را مشابه با سوال های قبلی، Load کرده و سپس دو محصول را به عنوان template برای آن قرار دادیم.

عینا مشابه با بخش اول سوال قبلی، محصولات را پیدا کردیم، دور آنها مستطیل کشیده و در تصاویر مجزا نمایش دادیم. این بار اما باید تعداد را هم گزارش می کردیم.

این روش از template matching چندین نقطه کنار هم را detect می کند و مستطیل هایی را روی هم می کشد که فاصله اندکی از هم دارند و به چشم نمی آید. چون در تصویر ما تمام محصولات مشابه کنار هم در یک ردیف بودند، پس min و max مقادیر ذخیره شده در آرایه Loc را یافته، از هم کم کرده و بر عرض تصویر template تقسیم کرده، سپس آن را به بالا رند کردیم تا تعداد محصولات هم مشخص شود.

نتایج هر بخش در فایل تحت عنوان Q3\_HW2\_9923069 قابل مشاهده است.

