

: Blob

Blob (به معنای "نقطه‌ای متصل و یکنواخت در تصویر") به ناحیه‌ای از تصویر گفته می‌شود که در آن، ویژگی‌های تصویری مانند شدت روشنایی، رنگ یا بافت به طور نسبتاً یکنواخت هستند و از نواحی اطرافشان قابل تفکیک می‌باشند Blob معمولاً نمایانگر یک شیء یا بخشی از شیء در تصویر است و می‌تواند در اندازه، شکل و شدت متفاوت باشد.

بنابراین:

Blob در OpenCV به ناحیه‌هایی از تصویر گفته می‌شود که خصوصیات تصویری یکسانی دارند، مثل:

- شدت روشنایی ثابت
- رنگ یکسان یا مشابه
- شکل مشخص (مثل دایره‌ای، بیضی شکل)
- اندازه‌ی مشخصی در تصویر

این نواحی معمولاً به صورت نقاط، دایره‌ها یا نواحی پیوسته ای دیده می‌شوند که به‌طور خودکار در تصویر شناسایی می‌شوند.

OpenCV برای تشخیص Blob یک کلاس به نام **cv2.SimpleBlobDetector** دارد.

این ابزار نقاطی را پیدا می‌کند که ویژگی‌هایی مطابق با تنظیمات ما دارند، مثلاً:

- حداقل و حداکثر اندازه‌ی Blob
- اینکه Blob باید تیره یا روشن باشد
- اینکه Blob دایره‌ای، محدب یا مقعر باشد

خروجی تابع `detector.detect()` لیستی از ****KeyPoint**** ها است که برای هر Blob این اطلاعات را بر می‌گرداند:

- مختصات مرکز (x, y)
- اندازه قطر blob
- زاویه (در صورت لزوم)

برای تشخیص blob در یک تصویر از 2 تابع اصلی در کتابخانه open cv استفاده می شود:

```
detector = cv2.SimpleBlobDetector_create()
```

• این تابع، یک شیء از نوع **SimpleBlobDetector** می سازد.

• وظیفه‌ی این شیء، شناسایی نواحی متصل در تصویر است که ویژگی‌های خاصی دارند، مثل اندازه، گردی، تیرگی یا روشنی. در صورتی که پارامتر خاصی برای این شی در نظر گرفته نشود از پارامترهای پیش فرض برای آن استفاده می شود.

```
keypoints = detector.detect(img)
```

- با استفاده از شیء **detector**، تصویر **img** را بررسی کرده و نقاطی که به عنوان **blob** شناسایی می شوند را تشخیص می دهد.
- خروجی این خط یک لیست از **cv2.KeyPoint** است، که هر **KeyPoint** نشان دهنده‌ی یک **Blob** در تصویر است.

هر **KeyPoint** چه اطلاعاتی دارد؟ 🧠

هر **blob** شناسایی شده، به شکل یک شیء **KeyPoint** ذخیره می شود که شامل موارد زیر است:

ویژگی	توضیح
pt	مختصات مرکز blob (x, y)
size	قطر blob بر اساس مساحت
angle	زاویه blob در صورت نیاز
response	میزان اطمینان الگوریتم به این نقطه
octave	مقیاس تصویری که blob در آن یافت شده
class_id	شناسه‌ی دلخواه (برای دسته بندی)

در نهایت با استفاده از کد زیر می توانیم blob ها را رسم کنیم:

```
result2 = cv2.drawKeypoints(img, keypoints, np.array([]), (0, 0, 255),
cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
```

در این خط، ابتدا از تابع `cv2.drawKeypoints` استفاده شده که مخصوص رسم نقاط کلیدی (KeyPoints) روی یک تصویر است. در اینجا، این نقاط کلیدی همان blob هایی هستند که قبلاً با استفاده از آشکارساز Blob شناسایی شده اند.

پارامتر اول این تابع، یعنی `img`، تصویر اصلی است که می خواهیم blob ها را روی آن رسم کنیم. پارامتر دوم، یعنی `keypoints`، لیستی از blob های شناسایی شده است که هر کدام شامل اطلاعاتی مانند مکان (مختصات مرکز) و اندازه blob می باشند.

پارامتر سوم `np.array([])`، یک آرایه ی خالی است که در نسخه های قدیمی تر OpenCV به عنوان تصویر خروجی مورد استفاده قرار می گرفت. امروزه نیازی به مقداردهی خاصی برای آن نیست و معمولاً همان آرایه خالی داده می شود.

پارامتر چهارم `(0, 0, 255)`، رنگ رسم blob ها را مشخص می کند. این مقدار در فضای رنگی BGR است نه RGB، بنابراین رنگ blob ها قرمز خواهد بود.

پارامتر پنجم `cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS` یک فگ (پرچم) خاص است که مشخص می کند هر blob باید به صورت کامل تری رسم شود. به این معنا که نه فقط مرکز blob، بلکه اندازه ی آن هم به صورت یک دایره دور نقطه نمایش داده شود. اگر این فگ استفاده نشود، فقط یک نقطه کوچک به عنوان مرکز blob رسم می شود.

لیست فگ های قابل استفاده در `cv2.drawKeypoints()` ✓

1. `cv2.DRAW_MATCHES_FLAGS_DEFAULT`

- توضیح: حالت پیش فرض. فقط مرکز هر keypoint (blob) به صورت یک دایره کوچک کشیده می شود.
- نمایش اندازه ✗ blob:
- نمایش جهت ✗ blob:

2. cv2.DRAW_MATCHES_FLAGS_DRAW_OVER_OUTIMG

- **توضیح:** به جای اینکه تصویر خروجی جدید تولید شود، blob ها را مستقیماً روی تصویر اصلی رسم می کند.
 - **کاربرد:** وقتی بخواهی تغییرات را روی خود img اعمال کنی.
 - **نکته:** باید مقدار پارامتر outImage را برابر با خود تصویر اصلی قرار دهی.
-

3. cv2.DRAW_MATCHES_FLAGS_NOT_DRAW_SINGLE_POINTS

- **توضیح:** فقط نقاطی را رسم می کند که **matching** یا تطبیق داشته باشند. نقاط تکی که به هیچ نقطه دیگری مرتبط نیستند، رسم نمی شوند.
 - **کاربرد اصلی:** بیشتر برای تطبیق بین دو تصویر استفاده می شود، نه فقط Blob.
-

4. cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS

- **توضیح:** نقاط را با جزئیات کامل رسم می کند، یعنی:
 - دایره ای به اندازه قطر blob
 - جهت آن (در صورت وجود)
- **پرکاربردترین حالت برای Blob ها است.**