توابع مهم ریاضی در opencv

۱. تابع cv2.cvtColor

این تابع برای تبدیل فضای رنگی یک تصویر به فضای رنگی دیگر استفاده می‌شود.

ورودی‌ها:

src: تصویر ورودی که می‌تواند یک تصویر رنگی (BGR) یا خاکستری باشد.

code: کد نوع تبدیل فضای رنگی. برخی از کدهای رایج:

cv2.COLOR_BGR2GRAY: تبدیل تصویر از فضای رنگی BGR به خاکستری.

cv2.COLOR_BGR2HSV: تبدیل تصویر از فضای BGR به HSV (رنگ، اشباع، روشنایی).

خروجی:

تصویر تبدیل‌شده در فضای رنگی جدید.

مثال:

```
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

این خط تصویر ورودی را به تصویر خاکستری تبدیل می‌کند.

۲. تابع cv2.threshold

این تابع برای آستانه‌گذاری (Thresholding) استفاده می‌شود، که در آن مقادیر پیکسل‌ها به مقدار مشخصی تبدیل می‌شوند. معمولاً در پردازش تصویر برای جداسازی اشیاء از پس‌زمینه کاربرد دارد.

ورودی‌ها:

src: تصویر ورودی (باید خاکستری باشد).

thresh: مقدار آستانه. هر پیکسل با مقدار بیشتر یا کمتر از این مقدار، تغییر داده می‌شود.

maxval: مقدار حداکثری که به پیکسل‌هایی که شرایط را دارند، اختصاص داده می‌شود.

type: نوع آستانه‌گذاری. برخی از حالت‌های رایج:

cv2.THRESH_BINARY: اگر مقدار پیکسل بزرگ‌تر از آستانه باشد، برابر با maxval می‌شود؛ در غیر این صورت صفر.

cv2.THRESH_BINARY_INV: معکوس حالت THRESH_BINARY.

cv2.THRESH_TRUNC: اگر مقدار پیکسل بزرگ‌تر از آستانه باشد، برابر با مقدار آستانه می‌شود؛ در غیر این صورت تغییری نمی‌کند.

خروجی:

یک عدد که مقدار آستانه محاسبه‌شده است.

تصویر آستانه‌گذاری شده.

مثال:

```
binary_image = cv2.threshold(gray_image, 127, 255, cv2.THRESH_BINARY)
```

این خط تمامی پیکسل‌هایی که مقدارشان بیشتر از ۱۲۷ است را به ۲۵۵ و بقیه را به ۰ تبدیل می‌کند.

۳. تابع cv2.resize

این تابع برای تغییر اندازه تصویر استفاده می‌شود. می‌توانید تصویر را کوچک‌تر یا بزرگ‌تر کنید.

ورودی‌ها:

src: تصویر ورودی.

dsize: اندازه نهایی تصویر به صورت (عرض، ارتفاع).

fx و fy (اختیاری): مقیاس برای تغییر اندازه تصویر در جهت افقی و عمودی. اگر dsize مشخص شود، این مقیاس‌ها نادیده گرفته می‌شوند.

interpolation: روش استفاده‌شده برای بازنمونه‌گیری (resampling). برخی روش‌های معمول:

cv2.INTER_NEAREST: ساده‌ترین روش.

cv2.INTER_LINEAR: مناسب برای بزرگ‌کردن تصاویر.

cv2.INTER_AREA: مناسب برای کوچک‌کردن تصاویر.

خروجی:

تصویر تغییر اندازه داده‌شده.

مثال:

```
resized_image = cv2.resize(image, (100, 200))
```

این خط اندازه تصویر را به عرض ۱۰۰ و ارتفاع ۲۰۰ تغییر می‌دهد.

۴. تابع cv2.getRotationMatrix2D

این تابع یک ماتریس برای چرخش دوبعدی تصویر ایجاد می‌کند.

ورودی‌ها:

center: نقطه مرکزی چرخش به صورت (x, y).

angle: زاویه چرخش به درجه. مقدار مثبت برای چرخش پادساعتگرد و مقدار منفی برای چرخش ساعتگرد.

scale: مقیاس تصویر پس از چرخش.

خروجی:

ماتریس چرخش دوبعدی.

مثال:

```
rotation_matrix = cv2.getRotationMatrix2D((cols / 2, rows / 2), 45, 1)
```

این خط ماتریسی ایجاد می‌کند که تصویر را ۴۵ درجه حول مرکز تصویر بدون تغییر اندازه بچرخاند.

۵. تابع cv2.warpAffine

این تابع برای اعمال یک تبدیل خطی (Affine Transformation) به تصویر استفاده می‌شود. تبدیل Affine شامل چرخش، برش، ترجمه و مقیاس‌دهی است.

ورودی‌ها:

src: تصویر ورودی.

M: ماتریس تبدیل دوبعدی (۳x۲).

dsize: اندازه تصویر خروجی.

خروجی:

تصویر تبدیل‌شده.

مثال:

```
rotated_image = cv2.warpAffine(image, rotation_matrix, (cols, rows))
```

این خط چرخش تعریف‌شده توسط ماتریس rotation_matrix را روی تصویر اعمال می‌کند.

۶. تابع cv2.getAffineTransform

این تابع برای محاسبه ماتریس تبدیل Affine استفاده می‌شود.

ورودی‌ها:

src_points: سه نقطه از تصویر ورودی که مختصات‌شان مشخص است.

dst_points: سه نقطه از تصویر خروجی که مختصات‌شان مشخص است.

خروجی:

ماتریس تبدیل Affine.

مثال:

```
affine_matrix = cv2.getAffineTransform(pts1, pts2)
```

این خط ماتریسی ایجاد می‌کند که نقاط pts1 را به نقاط pts2 نگاشت کند.

۷. تعریف ترجمه با ماتریس

برای جابجایی (Translation) یک تصویر، از ماتریس انتقال استفاده می‌کنیم.

مثال:

```
translation_matrix = np.float32([[۱, ۰, ۱۰۰], [۰, ۱, ۵۰]])
translated_image = cv2.warpAffine(image, translation_matrix, (cols, rows))
```

این کد تصویر را ۱۰۰ پیکسل به سمت راست و ۵۰ پیکسل به سمت پایین جابجا می‌کند.