

خواندن تصاویر:

تصاویر در پایتون به صورت یک ماتریس خوانده می شوند که این ماتریس در open cv به صورت یک آرایه سه بعدی numpy نمایش داده می شود. بنابراین در open cv ما تصاویری را به صورت Data structure های numpy لود می کنیم. در ادامه با متد هایی که برای خواندن و نوشتن و ذخیره سازی عکس ها با آنها سروکار داریم را باهم بررسی می کنیم

متد imread :

cv2.imread یکی از مهم ترین توابع کتابخانه OpenCV است که برای خواندن تصاویر از فایل های مختلف به کار می رود. این تابع یک تصویر را به عنوان یک ماتریس چند بعدی (ndarray) در کتابخانه NumPy بارگذاری می کند که هر عنصر این ماتریس نشان دهنده شدت روشنایی یک پیکسل در تصویر است.

```
cv2.imread(filename, flags=cv2.IMREAD_COLOR)
```

متد imshow :

cv2.imshow یک تابع مهم در کتابخانه OpenCV است که برای نمایش تصاویر در یک پنجره استفاده می شود. به عبارت ساده تر، این تابع تصویری که شما پردازش کرده اید را در یک پنجره جداگانه نمایش می دهد تا بتوانید نتیجه کار خود را ببینید.

```
cv2.imshow(window_name, img)
```

متد waitkey :

تابع **cv2.waitKey()** در OpenCV یک تابع بسیار مهم برای کنترل نمایش تصاویر و ایجاد تعامل با کاربر است. این تابع به برنامه اجازه می دهد تا منتظر فشار دادن یک کلید توسط کاربر بماند و سپس بر اساس آن کلید، عملیات بعدی را انجام دهد. ما می توانیم به عنوان ورودی مقداری برای صبر کردن بدهیم که به صورت میلی ثانیه لحاظ می شود اما اگر مقداری داده نشود به معنی صفر است و تا زمانی که کلیدی فشرده نشود به خط بعد نمی رود.

```
cv2.waitKey(delay)
```

متد destroyAllWindows :

تابع `cv2.destroyAllWindows()` در کتابخانه **OpenCV** برای بستن تمامی پنجره‌های نمایش تصویر که توسط این کتابخانه ایجاد شده‌اند، استفاده می‌شود. به عبارت ساده‌تر، این تابع تمامی پنجره‌هایی که برای نمایش تصاویر از طریق **OpenCV** باز کرده‌اید را یکجا می‌بندد.

چرا از این تابع استفاده می‌کنیم؟

- مدیریت منابع: وقتی کار با تصاویر در **OpenCV** تمام می‌شود، بستن پنجره‌های نمایش تصویر باعث آزاد شدن منابع سیستم و جلوگیری از مشکلات احتمالی می‌شود.
- پاک‌سازی محیط کار: با بستن تمامی پنجره‌ها، محیط کار شما مرتب‌تر و آماده برای اجرای کدهای جدید می‌شود.
- جلوگیری از همپوشانی پنجره‌ها: در برنامه‌هایی که چندین پنجره نمایش تصویر ایجاد می‌کنند، این تابع باعث می‌شود که پنجره‌های قدیمی بسته شده و تنها پنجره‌های جدید نمایش داده شوند.

بررسی type یک عکس:

در **open cv** با استفاده از دستور زیر می‌توان با تایپ یک عکس آشنا شد:

```
print(img.shape)
```

پس از اجرای این کد تایپ عکس به صورت ارتفاع در عرض در تعداد کانال‌های رنگی نمایش داده می‌شود. و دلیل این امر خوانده شدن تصویر به صورت ماتریس می‌باشد.

متد imwrite :

تابع `cv2.imwrite()` در کتابخانه **OpenCV** برای ذخیره تصاویر پردازش شده یا ایجاد شده به عنوان یک فایل تصویری استفاده می‌شود. به عبارت دیگر، این تابع به شما اجازه می‌دهد تا نتایج پردازش تصویر خود را به عنوان یک تصویر جدید ذخیره کنید.

```
cv2.imwrite(filename, img)
```

ما می توانیم تصاویر خود را با استفاده از کتابخانه **matplotlib** نیز نمایش دهیم. چراکه در این کتابخانه نوعی نمودار به نام **heatmap** وجود دارد که می تواند با خواندن پیکسل های تصویر به نوعی عکس را برای ما نمایش دهد. با استفاده از دستور زیر می توان یک عکس را در **matplotlib** نمایش داد:

```
plt.imshow(img)
```

اما بعد از فراخوانی این متد با عکسی عجیب روبرو می شویم که ترتیب کانال های آن به جای **RGB** به **BGR** تبدیل شده و ما باید به نحوی آن را به حالتی که برایشان فریم ورک ها آشناست تبدیل کنیم.

یکی از راه ها استفاده از **numpy** و پیمایش ترتیب کانال ها به صورت برعکس با استفاده از دستور زیر است:

```
plt.imshow(img[...::-1])
```

و راه دیگر با استفاده از متد **cvtColor** به شکل زیر است:

```
rgb_image = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

```
plt.imshow(rgb_image)
```

تابع **cv2.cvtColor** یکی از توابع بسیار کاربردی در کتابخانه **OpenCV** است که برای تبدیل تصاویر بین فضاهای رنگ مختلف به کار می رود. به عبارت ساده تر، این تابع به شما اجازه می دهد تا تصویر خود را از یک فضای رنگ (مثل **RGB** به فضای رنگ دیگری مثل **HSV** یا **Gray** تبدیل کنید.