ترسیم خطوط و اشکال هندسی با open cv :

OpenCVیک کتابخانه قدرتمند برای بینایی کامپیوتر و پردازش تصویر است که به شما امکان می دهد انواع مختلفی از اشکال هندسی را روی تصاویر ترسیم کنید. این اشکال می توانند شامل خطوط، مستطیل ها، دایره ها، بیضی ها، چند ضلعی ها و موارد دیگر باشند.

در ابتدا بیاید سه نوع تصویر رنگی، خاکستری و png را با هم ایجاد نماییم.

برای ایجاد یک تصویر رنگی می توان از قطعه کد زیر استفاده نمود:

```
color_image = np.zeros((500,500,3), np.uint8)
```

این کد یک تصویر خالی با ابعاد 500 500x500پیکسل و فرمت RGB ایجاد می کند.

color_image = np.zeros((500,500,3), np.uint8): این خط کد یک آرایه سه بعدی با ابعاد 500 500x500x3ایجاد می کند و آن را به عنوان color_imageذخیره می کند.

- np.zeros():این تابع یک آرایه پر از صفر ایجاد می کند.
- (500,500,3): این تاپل ابعاد آرایه را مشخص می کند. 500 سطر، 500 ستون و 3 کانال (قرمز، سبز، آبی).
- np.uint8:این نوع داده نشان می دهد که هر عنصر آرایه یک عدد صحیح 8 بیتی بدون علامت است. این نوع داده معمولا برای نمایش رنگ ها در تصاویر استفاده می شود.

برای ایجاد یک تصویر با سطح خاکستری می توان از کد زیر استفاده نمود:

```
gray_image = np.zeros((500,500), np.uint8)
```

این کد یک تصویر خالی با ابعاد 500 500x500پیکسل ایجاد می کند که به صورت سیاه و سفید (grayscale) نمایش داده می شود. در اینجا توضیح گام به گام این کد آمده است:

- **gray_image = ...**:این بخش یک متغیر به نام gray_imageتعریف می کند که تصویر ایجاد شده در این کد در آن ذخیره می شود.
- **np.zeros((500,500), np.uint8)**:این بخش با استفاده از کتابخانه NumPy یک آرایه دو بعدی از صفرها ایجاد می کند. این آرایه در واقع همان تصویر ما خواهد بود .
  - **np.zeros()**:این تابع آرایه ای از صفرها ایجاد می کند.

- **(500,500)**: این تاپل ابعاد آرایه را مشخص می کند که در اینجا 500 سطر و 500 ستون است. این یعنی تصویر ما 500 پیکسل عرض و 500 پیکسل ارتفاع خواهد داشت.
- **np.uint8**: این نوع داده، نوع داده عناصر آرایه را مشخص می کند `uint8`. مخفف "unsigned 8-bit integer" است و به این معناست که هر عنصر آرایه می تواند یک عدد صحیح بین 0 تا 255 باشد. این محدوده اعداد برای نمایش میزان روشنایی یک پیکسل در تصویر grayscale مناسب است.

و در نهایت برای ساخت یک تصویر 4 کاناله می توان از کد زیر استفاده نمود:

```
png_image = np.zeros((500,500,4), np.uint8)
```

این کد یک تصویر خالی با فرمت PNG در کتابخانه OpenCV پایتون ایجاد می‌کند. در اینجا توضیح گام به گام کد آمده است:

- **import numpy as np**: این خط، کتابخانه NumPy را وارد می‌کند و به آن نام مستعار `np` می‌دهد. NumPy یک کتابخانه قدرتمند برای کار با آرایه‌های چند بعدی در پایتون است. در اینجا، از NumPy برای ایجاد آرایه representing تصویر استفاده می‌شود.
- **png_image = np.zeros((500,500,4), np.uint8)**: این خط، یک آرایه NumPy با ابعاد 500 x500x4 ایجاد می‌کند و آن را به متغیر `png_image` اختصاص می‌دهد .
  - **(500,500,4)**: این تاپل، ابعاد آرایه را مشخص می‌کند. 500 x500 نشان دهنده عرض و ارتفاع تصویر به پیکسل است. 4 نشان دهنده تعداد کانال‌های رنگی است. از آنجا که فرمت PNG از کانال آلفا (شفافیت) پشتیبانی می‌کند، 4 کانال (قرمز، سبز، آبی و آلفا) در نظر گرفته شده است.
  - **np.uint8**: این بخش، نوع داده عناصر آرایه را مشخص می‌کند `uint8`. به معنای اعداد صحیح بدون علامت 8 بیتی است که مقداری بین 0 تا 255 دارند. هر کانال رنگی (قرمز، سبز، آبی و آلفا) با یک عدد 8 بیتی نمایش داده می‌شود.
  - **np.zeros(...)**: این تابع، یک آرایه با ابعاد مشخص شده و با مقادیر صفر ایجاد می‌کند. از آنجا که تصویر ابتدا خالی است، همه پیکسل‌ها با مقدار صفر مقداردهی می‌شوند.

نحوه رسم خط:

جهت رسم یک می توان از کد زیر استفاده نمود:

```
cv2.line(image, starting cordinates, ending cordinates, color, thickness)
```

`cv2.line()`: این تابع OpenCV است که یک خط را روی یک تصویر ترسیم می کند.

`image`: این تصویر است که خط روی آن ترسیم می شود. این می تواند یک تصویر موجود باشد که با استفاده از `cv2.imread()`بارگذاری شده است، یا می تواند یک تصویر خالی باشد که با استفاده از `np.zeros()` ایجاد شده است.

`starting coordinates`: این مختصات نقطه شروع خط است. این به عنوان یک تاپل از دو عدد صحیح نشان داده می شود، که مختصات x و y هستند.

`ending coordinates`: این مختصات نقطه پایان خط است. این نیز به عنوان یک تاپل از دو عدد صحیح نشان داده می شود، که مختصات x و y هستند.

`color`: این رنگ خط است. این می تواند به عنوان یک تاپل از سه عدد صحیح نشان داده شود، که مقادیر قرمز، سبز و آبی هستند. به عنوان مثال، `(255, 0, 0)` (قرمز است)، `(0, 255, 0)` (سبز است)، `(0, 0, 255)` (آبی است).

`thickness`: این ضخامت خط است. این به عنوان یک عدد صحیح نشان داده می شود.

نحوه رسم مستطیل:

جهت رسم یک مستطیل در کتابخانه open cv می توان از کد زیر استفاده نمود:

```
cv2.rectangle(image, starting vertex, opposite vertex, color, thickness)
```

این کد از کتابخانه OpenCV (Open Source Computer Vision Library) برای رسم مستطیل روی تصاویر استفاده می‌کند. در اینجا نحوه عملکرد هر پارامتر آمده است:

- `image`: تصویری که می‌خواهید مستطیل را روی آن رسم کنید.
- `start_point`: نقطه شروع مستطیل. این یک تاپل از دو مقدار است که مختصات x و y نقطه گوشه بالا سمت چپ مستطیل را نشان می‌دهد.
- `end_point`: نقطه پایان مستطیل. این نیز یک تاپل از دو مقدار است که مختصات x و y نقطه گوشه پایین سمت راست مستطیل را نشان می‌دهد.
- `color`: رنگ مستطیل. این یک تاپل از سه مقدار است که مقادیر RGB رنگ را نشان می‌دهد. به عنوان مثال، `(255, 0, 0)` (قرمز است)، `(0, 255, 0)` (سبز است)، `(0, 0, 255)` (آبی است).
- `thickness`: ضخامت خط مستطیل. اگر این مقدار -1 باشد، مستطیل پر می‌شود.

ترسیم دایره:

جهت رسم یک دایره می توان از کد زیر استفاده نمود:

```
cv2.circle(image, center, radius, color, thickness)
```

این کد یک دایره روی تصویر رسم می کند.

- `Image` تصویری که می خواهید دایره را روی آن رسم کنید.
- `Center` مختصات مرکز دایره.
- `Radius` شعاع دایره.
- `Color` رنگ دایره.
- `Thickness` ضخامت خط دایره.

ترسیم بیضی:

جهت رسم یک دایره می توان از کد زیر استفاده نمود:

```
cv2.ellipse(image, center, radius, angle, startAngle, endAngle color, thickness)
```

`image` تصویری که می خواهید بیضی را روی آن بکشید
`center` مرکز بیضی (X,Y)
`radius` شعاع بیضی (محور اصلی، محور فرعی)
`angle` زاویه چرخش بیضی در جهت عقربه‌های ساعت
`startAngle` زاویه شروع قوس بیضی در جهت عقربه‌های ساعت
`endAngle` زاویه پایان قوس بیضی در جهت عقربه‌های ساعت
`color` رنگ بیضی
`thickness` ضخامت خط بیضی