**CÁC THƯ VIỆN THƯỜNG DÙNG TRONG THỊ GIÁC MÁY TÍNH**

Các thư viện phổ biến hiện thường được dùng trong thị giác máy tính như: OpenCV, Scipy, Tensor Flow, DNN, Numpy, Pillow, Os, BoofCV, CUDA,…. Nhưng thường được sử dụng nhiều nhất là 6 thư viện dưới đây:

1. DNN( deep neural network):

* là mô hình học sâu gồm nhiều tầng nơ-ron phức tạp
* được dùng để học với các đặc trưng phức tạp từ dữ liệu hình ảnh, âm thanh, văn bản
* dùng để : + nhận diện đối tượng

                + phân loại ảnh

                + phát hiện khuôn mặt, phân đoạn ảnh

| **Hàm** | **Chức năng** |
| --- | --- |
| cv2.dnn.readNetFromONNX(path) | Load mô hình ONNX |
| cv2.dnn.readNetFromCaffe(prototxt, caffemodel) | Load mô hình Caffe |
| cv2.dnn.readNetFromTensorflow(pb\_file) | Load TensorFlow |
| cv2.dnn.blobFromImage() | Biến ảnh thành blob (dạng tensor input cho model) |
| net.setInput(blob) | Gán input |
| net.forward() | Chạy mô hình và lấy output |

1. PILLOW

* mở, lưu, chuyển đổi định dạng ảnh( jpg, png,...)
* thao tác với ảnh: cắt, xoay, chỉnh màu,....
* Các bước thự hiện trong python: + from PIL import Image
* Mở và load ảnh trong thư mục hiện tại: img = Image.open("the\_catch.jpg")
* Xem thông tin của ảnh: image.show()

1. OpenCV

* Biến đổi hình học: xoay, scale, affine, perspective.
* Phát hiện cạnh: Canny, Sobel, Laplacian.
* Morphology: mở, đóng, giãn, co ảnh nhị phân.
* Tracking đối tượng (MeanShift, KCF, MOSSE,…).

| **Hàm** | **Chức năng** |
| --- | --- |
| cv2.imread(path) | Đọc ảnh từ file |
| cv2.imshow(name, image) | Hiển thị ảnh ra màn hình |
| cv2.waitKey(ms) | Đợi nhấn phím để đóng cửa sổ |
| cv2.cvtColor(img, code) | Chuyển đổi không gian màu (RGB ⇄ Gray ⇄ HSV …) |
| cv2.resize(img, size) | Resize ảnh theo kích thước mong muốn |
| cv2.Canny(img, threshold1, threshold2) | Phát hiện cạnh bằng thuật toán Canny |
| cv2.VideoCapture(0) | Mở webcam hoặc file video |
| cv2.imwrite(filename, img) | Lưu ảnh ra file |
| cv2.rectangle(img, pt1, pt2, color, thickness) | Vẽ hình chữ nhật lên ảnh |
| cv2.putText(img, text, pos, font, size, color, thickness) | Viết chữ lên ảnh |

1. Numpy

* **Lõi nền tảng** cho mọi hoạt động xử lý ảnh:
* Mỗi ảnh thực chất là 1 mảng ndarray 2D hoặc 3D.
* Dễ dàng thao tác từng kênh ảnh

| **Hàm** | **Chức năng** |
| --- | --- |
| np.array(object) | Tạo mảng NumPy từ list, ảnh, v.v. |
| np.zeros((h, w, c)) | Tạo ảnh đen (mảng 0) với kích thước cụ thể |
| np.ones((h, w)) \* 255 | Tạo ảnh trắng (giá trị 255) |
| np.mean(array) | Tính trung bình toàn bộ mảng |
| np.max(array), np.min(array) | Lấy giá trị lớn nhất / nhỏ nhất trong mảng |
| np.flip(array, axis) | Lật ảnh theo trục dọc hoặc ngang |
| np.reshape(array, new\_shape) | Đổi kích thước (reshape) mảng ảnh |
| array[:, :, 0] | Truy cập kênh màu riêng (0: R, 1: G, 2: B nếu dùng RGB) |
| array[50:100, 100:150] | Cắt vùng ảnh bằng slicing |
| np.where(condition) | Lọc phần tử thỏa mãn điều kiện (ví dụ: chọn pixel sáng hơn 200) |

1. OS

* Duyệt thư mục ảnh:
* Dùng để load dataset hoặc chia ảnh thành train/test.
* Có thể kết hợp với glob, shutil để quản lý file tốt hơn.
* Thường đi kèm với script chạy batch cho hàng trăm/thousands ảnh.

| **Hàm** | **Chức năng** |
| --- | --- |
| os.listdir(path) | Liệt kê tất cả file và thư mục trong path |
| os.path.join(a, b) | Ghép 2 đường dẫn lại đúng chuẩn hệ điều hành |
| os.path.exists(path) | Kiểm tra xem file hoặc thư mục có tồn tại không |
| os.makedirs(path) | Tạo thư mục (bao gồm cả thư mục cha nếu chưa có) |
| os.remove(file\_path) | Xóa file ở đường dẫn đã cho |
| os.rename(old\_name, new\_name) | Đổi tên file hoặc thư mục |
| os.path.splitext(filename) | Tách phần tên và phần đuôi mở rộng của file |
| os.getcwd() | Lấy đường dẫn thư mục hiện tại (current working directory) |
| os.chdir(path) | Thay đổi thư mục làm việc hiện tại |
| os.path.isfile(path) | Kiểm tra xem đường dẫn có phải là file không |

1. Scipy

* Mạnh về các **thuật toán xử lý tín hiệu** và ảnh số:
* Convolutions, làm mờ, phân ngưỡng ảnh, đo hình dạng.
* scipy.ndimage cực kỳ hữu ích trong xử lý ảnh grayscale.
* Thường dùng sau bước xử lý bằng NumPy hoặc chuyển ảnh từ OpenCV sang dạng float để xử lý chuẩn hơn.

| **Hàm** | **Chức năng** |
| --- | --- |
| ndimage.gaussian\_filter(image, sigma) | Làm mờ ảnh bằng bộ lọc Gaussian |
| ndimage.median\_filter(image, size) | Lọc nhiễu bằng Median |
| ndimage.sobel(image) | Phát hiện biên cạnh bằng Sobel |
| ndimage.rotate(image, angle) | Xoay ảnh một góc xác định |
| ndimage.zoom(image, zoom\_factor) | Phóng to hoặc thu nhỏ ảnh |
| ndimage.shift(image, shift=(x, y)) | Dịch chuyển ảnh theo trục x, y |
| ndimage.binary\_fill\_holes(image) | Lấp đầy các lỗ (hole) trong ảnh nhị phân |
| ndimage.label(image) | Gán nhãn các vùng liên thông trong ảnh nhị phân (segmentation cơ bản) |
| ndimage.measurements.center\_of\_mass(image) | Tính trọng tâm của vùng ảnh |