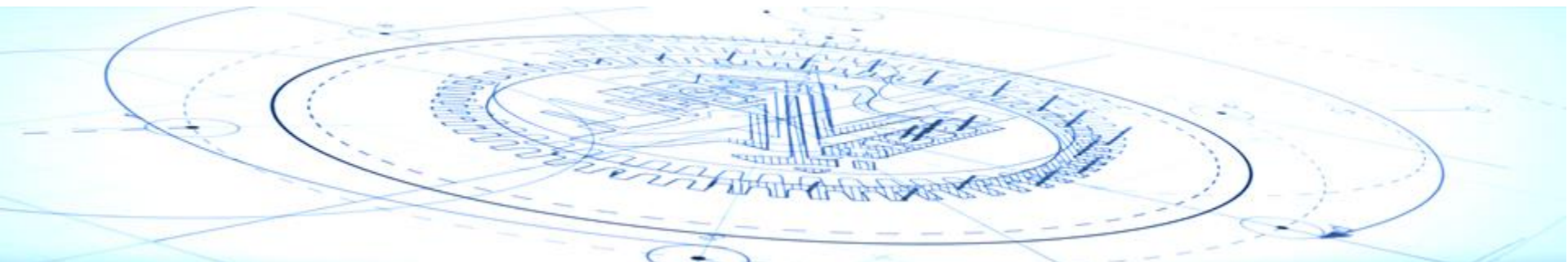


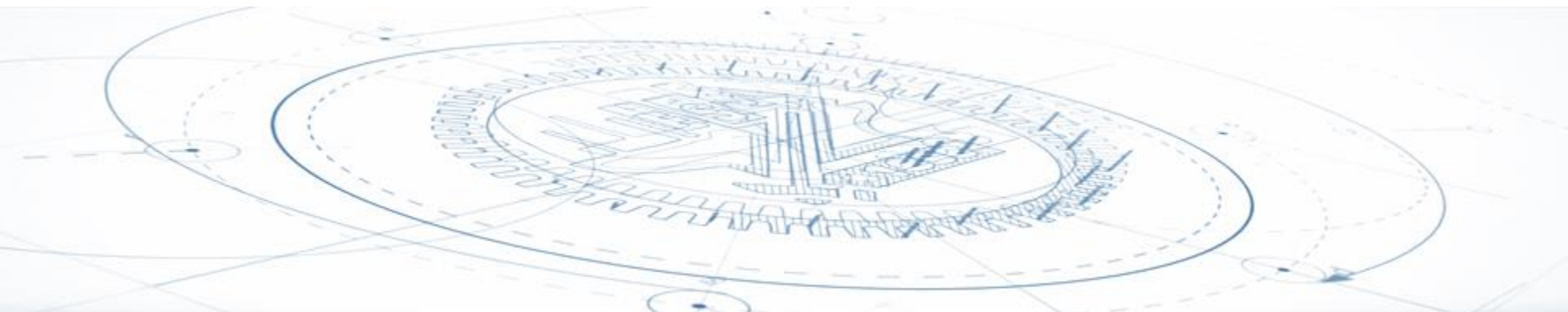
OOP 2025 Lab14

NumPy + OpenCV

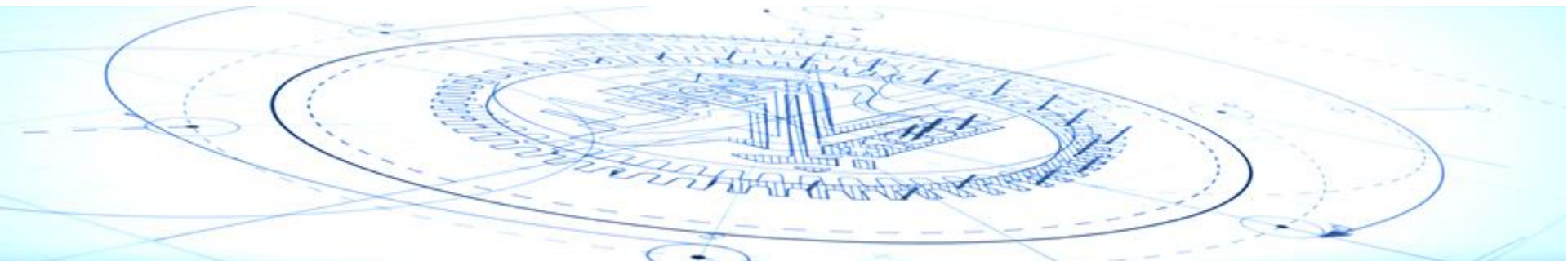


Outline

- NumPy intro
- OpenCV intro
- OpenCV task examples
- Lab



NumPy Intro



NumPy

What is NumPy?

A python library used for working with arrays.

Contains functions for linear algebra, fourier transform, matrices

More efficient than Lists \leq stored in a continuous place in memory

```
1 import numpy as np
2 arr = np.array([1,2,3,4,5])
3 print(arr)
4 print(type(arr))
```

✓ 0.1s

```
[1 2 3 4 5]
<class 'numpy.ndarray'>
```

```
1 a = np.array(42)
2 b = np.array([1, 2, 3, 4, 5])
3 c = np.array([[1, 2, 3], [4, 5, 6]])
4 d = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])
5 print(a.ndim)
6 print(b.ndim)
7 print(c.ndim)
8 print(d.ndim)
```

✓ 0.0s

```
0
1
2
3
```

```
1 import numpy as np
2
3 arr = np.array([1, 2, 3, 4, 5, 6, 7])
4 print(arr[4:])
5 print(arr[:4])
6 print(arr[1:6:2])
```

✓ 0.0s

```
[5 6 7]
[1 2 3 4]
[2 4 6]
```

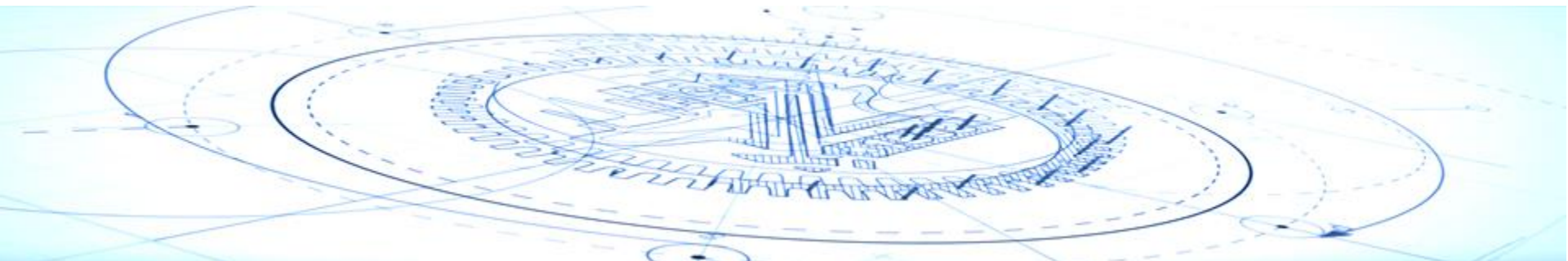
[And more](#)

```
1 import numpy as np
2
3 arr = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]])
4
5 print(arr[0:2, 1:4])
```

✓ 0.0s

```
[[2 3 4]
 [7 8 9]]
```

OpenCV Intro



OpenCV

What is OpenCV-python?

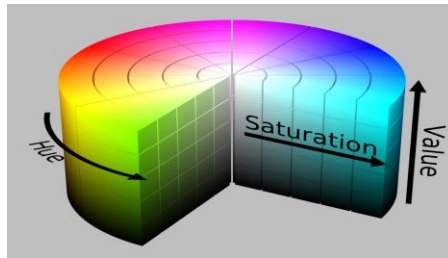
OpenCV-Python is a library of Python bindings designed to solve computer vision problems.

OpenCV-Python is a Python wrapper for the original OpenCV C++ implementation.

OpenCV-Python makes use of NumPy.

[And more](#)

OpenCV task examples



- Basic Tasks

Convert Image to grayscale

```
2 gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

Resize / Rotate

```
16 overlay_resized = cv2.resize(overlay, (w, h))
```

Draw shapes (lines, rectangles, circles)

```
7 cv2.rectangle(img_blobs, (x, y), (x+w, y+h), (255, 0, 0), 2)  
8 cv2.putText(img_blobs, f"ROI", (x, y-10), cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255, 0, 0), 2)
```

- Intermediate Tasks

Edge detection with Canny

```
9 eroded = cv2.erode(thresh, None, iterations=2)  
10 dilated = cv2.dilate(eroded, None, iterations=2)  
11 edges = cv2.Canny(dilated, 100, 200)
```

Color masking in HSV

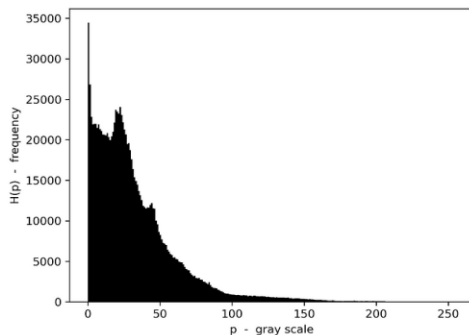
```
3 mask = cv2.inRange(hsv, (20, 100, 100), (40, 255, 255))
```

Contour detection

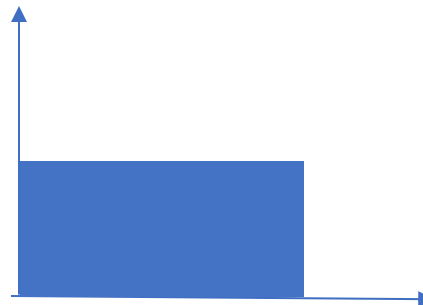
```
5 contours, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
```

- TA demo

Histogram Equalization



\Rightarrow



[for more](#)

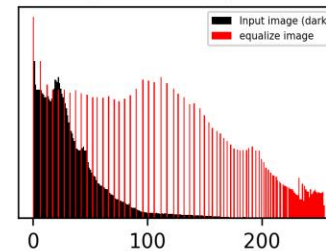
Input image (dark)



Equalized image (dark)



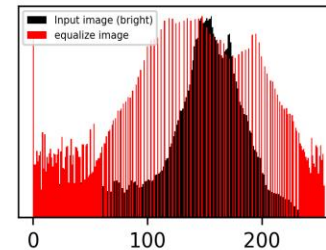
Grayscale comparison



Input image (bright)

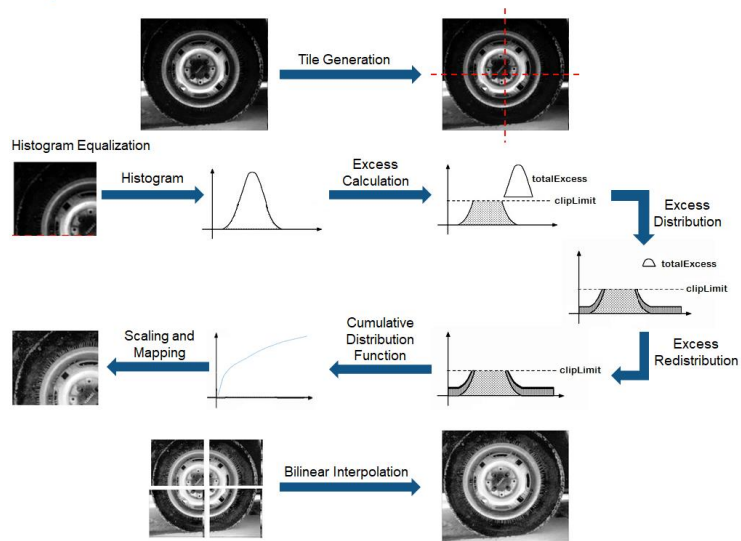


Equalized image (bright)



CLAHE (Contrast-Limited Adaptive Histogram Equalization)

CLAHE Algorithm



Original



Histogram Equalized



CLAHE Enhanced



[For more](#)

Lab

- Put the files on e3 under oop-python-nycu repo
- Demo 1: Change crying emoji into smiley emoji

Load image

Find Yellow contour

Bound ROI

Replace that crying face

- Demo 2: Do histogram equalization on provided image

Load image

Apply histogram equalization to whole image

Apply CLAHE (Contrast-Limited Adaptive Histogram Equalization) to the image

- Feel free to ask chat-GPT...