Section 1,2

Section 1 openCV - numpy Computer vision

① arr = numpy.array ([[1,2,3], [4,5,6]]) → 1,2,3 / 4,5,6

② arr = numpy.arrange (12) .reshape (3,4)
↳ random array with vales 3,4 (rows cols) with values → (0 → 11)

arr.size → كام قيمة
arr.shap → كام × كام
arr.item size → عدد القيم في كل صف
arr.ndim → كام dimension
arr.dtype.name → نوع الحاجة اللي جواها

③ np.zeros ((3,3), uint8) → مصفوفة 3x3 اصفار → Gray sc
(dtype)

④ img2 = cv2.cvtColor (img, cv2.COLOR_Gray2BGR)
(تحويل الالوان)

⑤ Check structure of an image by → Shape property
(rows, cols, number of Channels (if there is more than 1))

⑥ Imread (Filename, mode of image read) → load img from file

```
img = cv2.imread ('img')
cv2.imshow ("image", img)  →
cv2.waitKey (1000) → ms
cv2.destroyAllWindows ()
```

Mode of image read

CV2. ImRead _ (X)

① Color → default option → 3Channel BGR
② Grayscale → 8bit grayscale
③ Any Color → 3channel BGR or 8bit depedon metadata
④ unchanged → read all info of image data alpha, transp.
⑤ Any depth → gray scale + its original bit depth

⑥ Reduced_Grayscale_2 → reduce grayscale imge to 1/2 quality
⑦ Reduced _ Color _ 2 → " Colored " " " "

(4/8)

Image writing → Save image at specific place

CV. imwrite ('image.jpg'), var) → same directory
CV. Imwrite (r'location', var)

Convert between an image and raw Bytes
→ reshape

import CV, numpy, os

X→(bytearray (os.urandom (120 000))
Arr numpy. array (X) → vector

arr. reshape (300, 400) → 300 row 400 col Gray Scale

↓

(100, 400, 3) → BGr
100 row 400 col

/ /

videos

readvideo → VideoCapture

FPS → videoCaptue.get (cv.CAP_Prop_FPS)

size → " " " (cv.CAP_prop_FRAme_width),
" . " _ " _ " _ "height)

Section 2

Color models → Grayscale → 8bit → 0:255
→ BGR → 3x8bit → [x;y,z] 0:255
→ HSV uses diffrent triplet 3 Channel
Hue → tone Staturation → intensity
Value → brightness

Python → default → BGR

Color modes

① BGR to grayscale → cv2.cvtColor('original img', cv2.Color Bgr2Gray)

② Gray to Binary → (thresh, bwimage) = cv2.threshold('gray image', 127, 255, cv2.Thresh_binary)

① gray image
② threshold
③ value above threshold
④ CV method to convert Gray to binary

Scanned with CamScanner

Blue green red

red Green blue

③ BGR to RGB → cv2.Color_BGR toRGB

④ Blue only
red only
green only
} → B, G, R = cv2.split('img')
cv2.imshow(B)
" " (G)
" " (R)

⑤ merge B+G+R → Mergedimg = cv2.merge((B,G,R))

⑥ BGR to HSV
cv2
↳ .Color_BGR2HSV

⑦ upper and lower for specific Color

cv2.inrange(hsvimage, lower, upper)

array of lower and upper of specific Color

⑧ Resize image

as width, heigh not known

- half → cv2.resize(image, (0,0), fx=0.5, fy=0.5)
- biger → cv2.resize(image, (W,H))

- default of interpolation → Inter linear

Stretch = cv2.resize(image, (w, h), interpolation = cv2.InterNearest)

Low pass filtering (Blurring)

mask size

average filter → cv2.blur(image, (5,5))

Gaussian filter → cv2.GaussianBlur(image, (7,7), 2)

median → cv2.medianBlur(image, 5)

bilateral → cv2.bilateralFilter(image, 9, 75, 75)

Sigma Color

Sigma Space

Edge detection

- Custom
- Laplacian
- Sobel
- Scharr
- Canny