22/05/23.

Day 1

1) markdown in colab &
latex (for math eqns.)

2) learn x in y minutes → good python exercises.

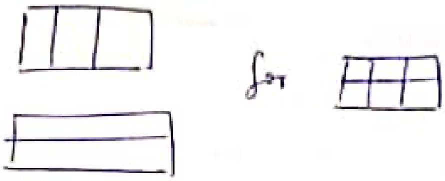
3) tuple vs lists
- immutability ⇒ frozenset([iterable])
  ↳ converts to immutable.

4) opencv works with ~~ma~~ numpy arrays.

5) numpy axis = 0

axis = 1

for



To do → (Github repo. Codes.) CV T Workshop Day X.
Session 1+2.

I. Images → Digital Image Processing.

→ $[x, y, z]$ $0 \leq z \leq 255$
↳ rgb

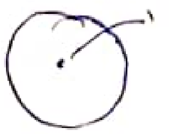
$[x, y]$ ↳ grayscale. $0 \leq x, y \leq 255$.

uint8
~~float int~~.
typecast to np.array.

cv2 assumes img is in uint8 →

$[x, y]$ b/w $0 \leq x, y \leq 1$

RGB, HSV → masking colours e.g.
↳ hue saturation value.



0-360°
↳ for colours.
spectrum.

Image is set of colours.

Ex Avg of rgb channel → grayscale.

bgR in opencv

iterate through bg cell.

James bond to circle => wherever fg == green ~~bg~~ bg cell

where fcell != green.

~~fg cell~~ bg cell - fg cell.

chromakeying.

II Image blurring → kernel (matrix to multipy with)

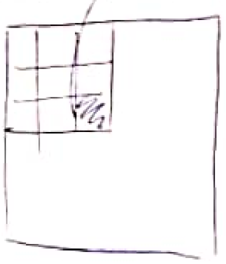
morphological processes.

Dilation + Erosion.

Binary mask → graryscale → b/w

↳ 0/1 values.

① Dilation → each cell is max of its 8 neighbours. } mask value.

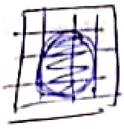


→ dilation of pixels.

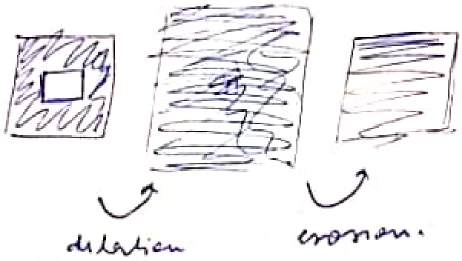
→ (enlarges sizes) ← here there is ② rectangular kernel

→ enlarges as rectangular.

① circular kernel



② Erosion → dilation calculates using min.



→ size same as original.

} closing operation,

more the size of mask ⇒ more blurring.

BARCODE SCANNING

① Assumption → largest part is the barcode.

② ceed fill algorithm to find largest area → recursively fill neighbouring cells

③ minimum rectangle which encloses barcode region.

closing operation → rank masks → longest area → minimum area rectangle ←

rectangle ⇓ bounding box. / quad optimization

→ perform additional dilation to make box not tightly bound

given ⇒ EAN 13 barcode.

say 6 661869 101132

1 2 K

↳ independent of 1/2.

||
ııı

↓ say check this in std. table 1 odd/even parity (xor of all). ~~0/1~~

say 6 :→ 1 2 2 2 1 1 for left. (1 is odd 2 is even)

so 661869 001100 draw a line.

(row by row) ⇒ hoping most common encoding is the correct one.

① first & last occurrence of 0 ⇒ i.e. 101 ka.

reqd. region.

② find no. of 0's that makes the 1 bar of 101 } assumption → standard for all bars.

③ Each EAN number is of form (7 bit).
xxxxxxx } where x = 0/1.

④ check with tables.
for first 6 → see parity
if == 1 ⇒ search left odd
if == 0 → search left even

next 6 ↳ store then find ~~[illegible]~~ 1st digit
then search right

Sobel filter → helps find dir. of gradient & gradient intensity.

Scanning + Detection.

Scanning ↓ difficult with normal method

Detection ↓ more standard.

[simple + global thresholding].

simple ↓ adaptive (decides threshold values based on small values).

global thresholding ↓ same for entire img.

THRESHOLDING IMP***