L-2 JAF250 20.9.22

Histogram is a 1D vector. It is part of point operation.

grb2grey → ~~sm~~ changes the value from 1 to 255 to (0 to 1)

full-had 3 dim. (X, Y, 3) — R G B

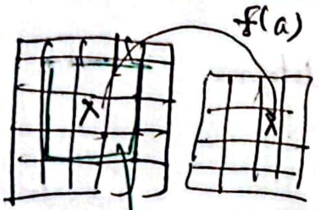
R X G Y B → (0, 1, 2)

S-4
- ছবি white, hist-এর ডানদিকে বেশি হবে → over exposed
  mean বাড়ানো: Time axis = ২ তে choose করবে।
- ছবি black, hist এর বাম দিকে বেশি হবে → under exposed.
  histogram will come → ~~manmalde~~ man-made.

S-17 skewness : measure for distribution / symmetry of hist

Kurtosis : measure of flatness / peakness

P-20



নিচে ৪ জায়গা আছে
pixel এর উপর depend
করে pixel (centre) update
হবে

Project (Instead of exam)
- MON → ques দিবে
- FRI → submission

last week of class

→ grade A to F

project with report

1-Day of work

→ but will give 5 day

L-2 JNF250 20.9.22

Histogram is a 1D vector. It is part of point operation.

rgb2grey → changes the value from 1 to 255 to (0 to 1)

fall - had 3 dim. (X, Y, 3) R G B

S-4 যদি white, hist-এর ডান দিকে বেশি হবে → over exposed
mean বাড়লে Time axis = 2 তে choose করবে (0, 1, 2) R, G, B

যদি black, hist এর বাম দিকে বেশি হবে → under exposed
histogram will cover → man-made.

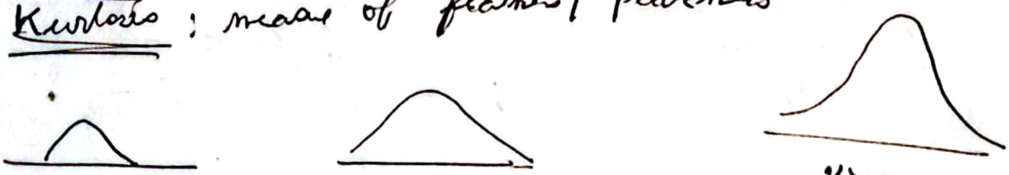
S-17 **skewness** : measure for distribution / symmetry of hist

negative <0 left

positive >0 right

normal distribution = 0

Kurtosis : measure of flatness / peakness

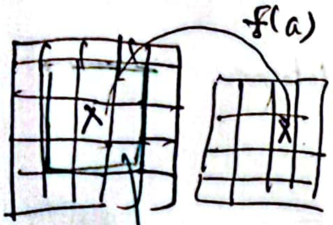


<0 low platy-

K = 0 normal distribution

K > 0 Lepto

P-20



f(a)

নিচের 3 জায়গা জুড়ে pixel এর উপর depend করে center pixel update হবে

Project (Instead of exam)
MON → ques দিবে
FRI → submission
last week of classes
→ grade A to F
project with report
1 Day of work
→ but will give 5 day

S-25 **THRESHOLDING** → for this histogram will be just 2 bars.

$$f_{threshold}(a) = \begin{cases} a_0 & \text{for } a < a_{th} \\ a_1 & \text{for } a \geq a_{th} \end{cases}$$

also - very common → HW 1

P-31 by modifying the histogram → you can increase the contrast.

→ useful for low quality images

P-32 Automatic contrast. → make the lowest to 0.] stretching
make the highest to 255] linear transformation

P-34 Modified Automatic contrast adjustment.

low - 0, highest - 256. saturated pixels → সেগুলোকে count করবে না → in min & high determination.

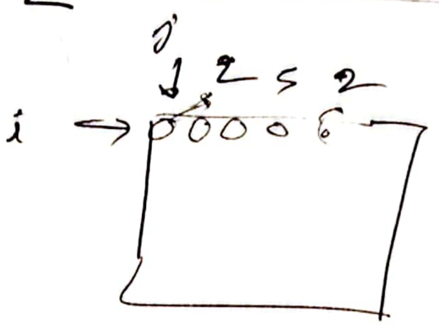
P-35 camera sees exactly the same light
human's see linearly.
↳ মানুষ কম extreme low & extreme high light এ কাজ করে না।

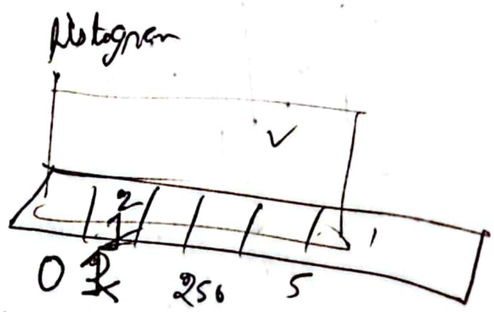
LAB - 3

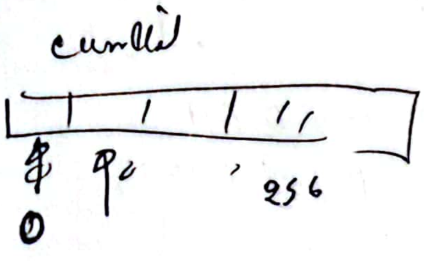
axis:

0 = col

1 = row







S-25 THRESHOLDING → for this histogram will be just 2 bars.

$$f_{threshold}(a) = \begin{cases} a_0 & \text{for } a < a_{th} \\ a_1 & \text{for } a \geq a_{th} \end{cases}$$

also - very common → HW 1

P-31 by modifying the histogram → you can increase the contrast.

→ useful for low quality images

P-32 Automatic contrast → make the lowest to 0. make the highest to 255] stretching / linear transformation

P-34 Modified Automatic contrast adjustment.

low - 0, highest - 255. saturated pixels → (দেখাবো)

count করে না → in max & high determination.

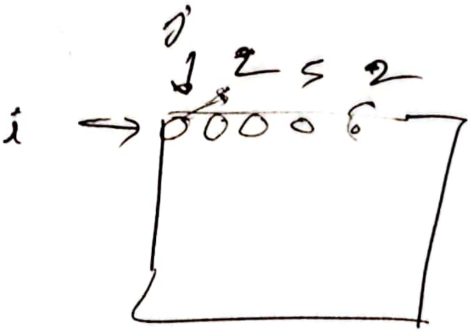
P-35 camera sees exactly the same light, human sees linearly.
↳ আমরা চোখ extreme low & extreme high light এ দেখতে পাই না।

LAB - 3

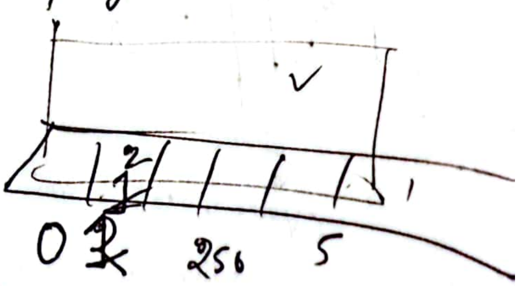
axis:

0 = col

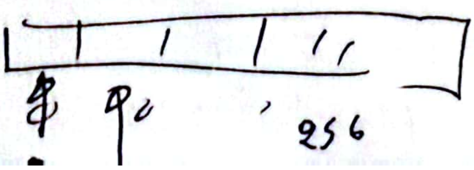
1 = row



histogram



cumulative

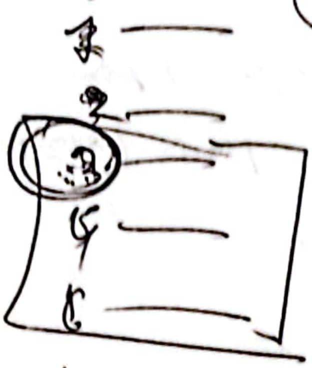


len(

N=3

LAB 3

6 − 3 = 3+1 = 7 5, 6, 7



will smooth
is reduc…

N=10

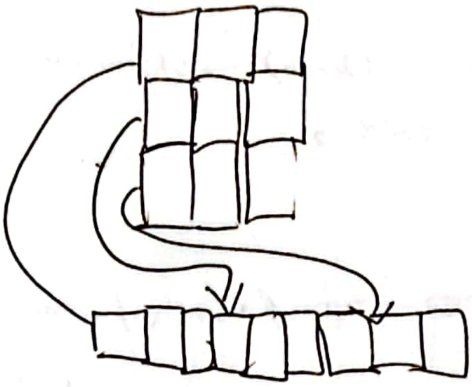
start
length

10 − 3 = 7

LOOP (3) — 8 9 10

step 7

flatten



sk image.
equalist_hist

plt.subplot(1, 1, figsize (13, 5))

big image

skimage.exposure.
~~skitlz~~ skitlz.
equalize_hist( )

MAX $\frac{30}{\cancel{255}}$

5

5

25

−5 + 30 = 25

bin_edges[0:-1] 4 6 original

plt.plot(edges, histogram)

27. 9. 22

L-3 (Filter)

Point operation → all operation is done for 1 pixel.

filter operation → works on several pixels. (3×3 pixel)

green → most important. (window size).

smoothing → mean filter

filter → 2 type: linear, non-linear.

Linear filter:

duplicate

process - filter - convolve (linear) → 5×5 → window

→ it will give edges with 24.

Gaussian blur:

more pixel → বেশি আসে → তবে of blurring / smoothing value.

linear filter

① → commutativity : I - image, H → filter.

② → linearity → scaler numbers.

③ associativity → B, C → filters.

④ separability (X/Y)

$H_x = [1\ 1\ 1]$

$H_y = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

$H_{xy} = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

blurred image does not have the edges. So when you subtract it with the original you get the edges

Analyzing residual image:

After blurring contrast points are gone.

image calculator → operation → subtract.

for enhancing image → also histogram.

edge detection

residual = filtered image (blurred image) - original image = will give blur edges

→ use equalize histogram for getting the proper edges

27. 9. 22

L-3 (Filter)

Point operation → all operation is done for 1 pixel.

filter operation → works on several pixels. (3×2 pixel)

Green → most important. (Window size).

smoothing → mean filter

filter → 2 type: linear, non-linear.

Linear filter:

duplicate

process - filter - convolve (linear) → 5×5 → window

→ it will give edges with 24.

Gaussian blur:

more pixel → নিবো যত → তত বাড়ে of blurring / smoothing value.

Linear filter

1. → commutativity : I - image, H → filter.
2. → linearity → scalar numbers.
3. , associativity → B, C → filters.
4. separability (X/Y)

$H_x = [1\ 1\ 1]$

$H_y = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

$H_{xy} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

blurred image does not have the edges. So when you subtract it with the original you get the edges

Analyzing residual image:

After blurring contrast points are gone.

image calculator → operation → subtract.

(for enhancing image → also histogram.

edge detection

- residual = filtered image (blurred image) − original image = will give blur edges
- → use equalize histogram for getting the proper edges

Non-linear filter :

mean - linear
median → non-linear.

Linear filter will smooth the borders. (noise reduce...
non-linear → does not

Border:

non-linear filter (non linear function → maximum, minimum filter
↳ better

salt-pepper → 0 & 255 add.

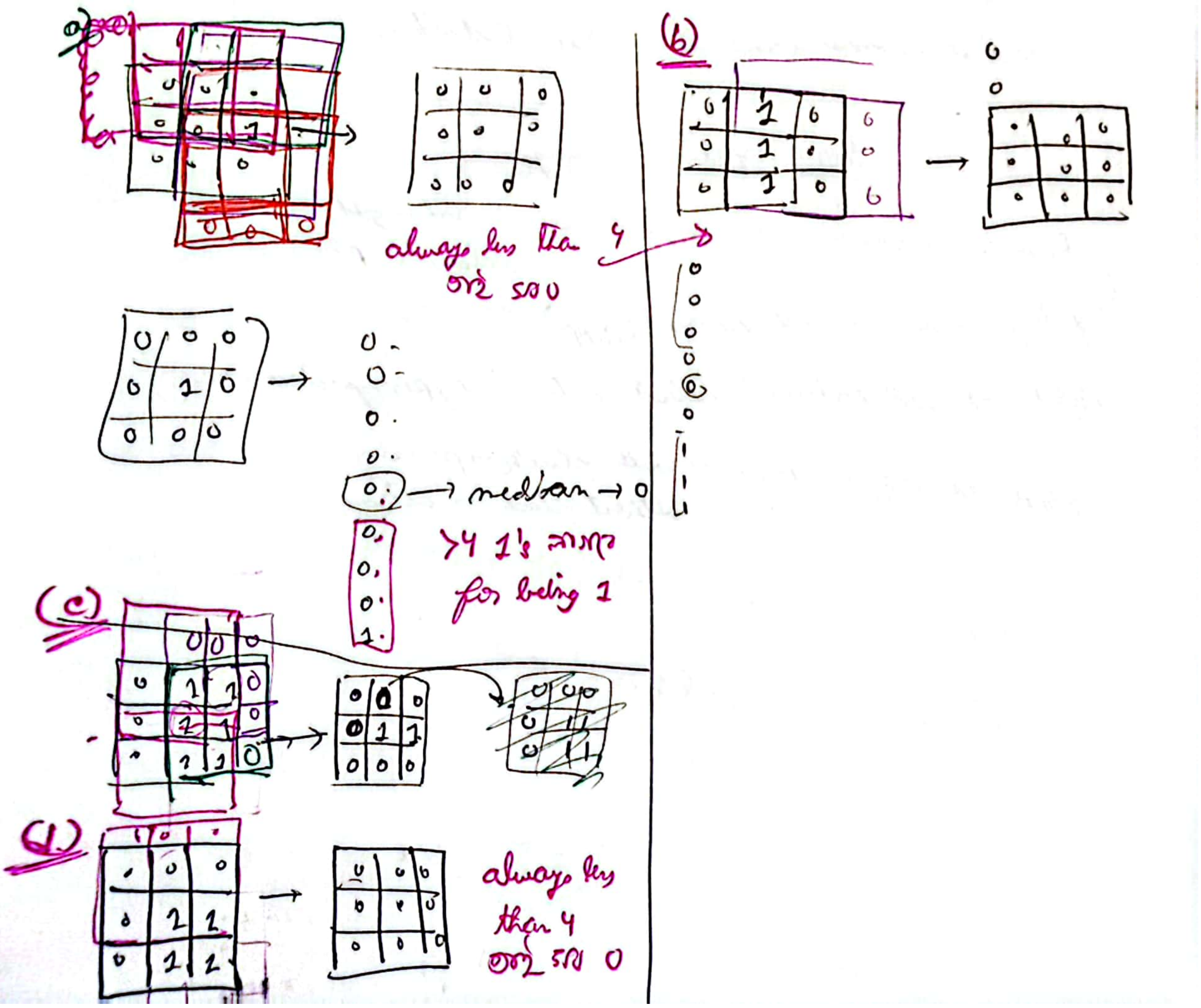
mean of 11245

$\frac{5/13/2°C \quad 10}{30}$

1 1) ② (4 5

non-linear median filter → does not smooth

better image
↳ add noise → add salt & pepper. → use median filter for smoothing & removing noise.



Non-linear filter:

mean - linear
median → non-linear.

linear filter will smooth the borders. (noise reduc...
non-linear → does not.

Border:

non-linear filter (non linear function → maximum, min/median filter)
→ better

salt-pepper → 0 & 255 add.

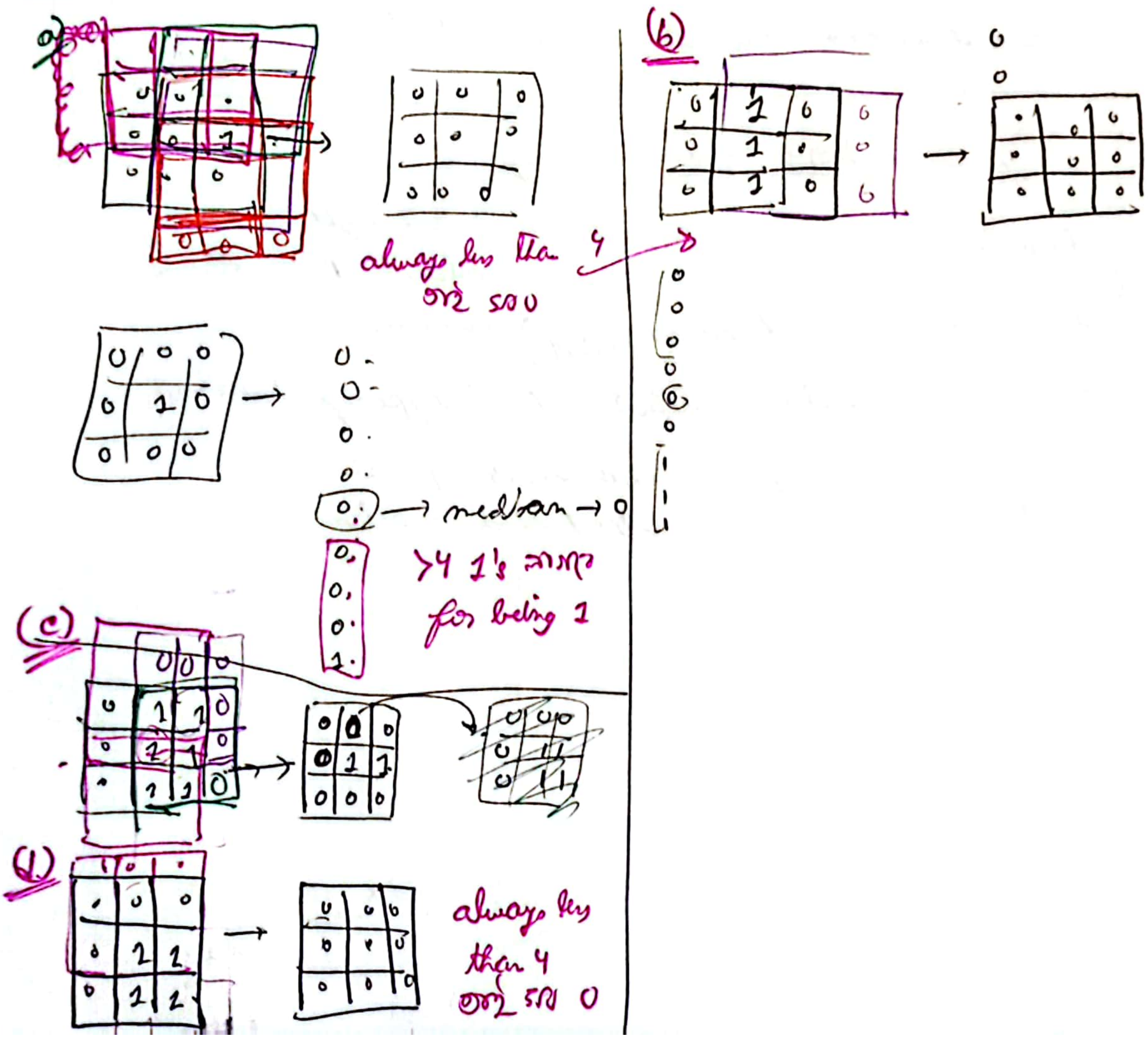
mean of 1 1 2 4 5

1 1) 2 (4 5

5/13/2°C
10
30

non-linear median filter → does not smooth

better image
→ add noise → add salt & pepper. → use median filter for smoothing & removing noise.



Non-local means :→ important for finding patterns in data

repetition → some similarity.

average neighboring → images

always same pixel, but different noises.

যত বেশির average, তত বেশি noise same হবে, — same, noise different

↪ reduce noise by average.

noise $= N^2$

$P_1 = \boxed{P_0} +$ noise → always random.

$P_2 = P_0 + n_2$

$P_N = P_0 + n_n$

always pixel will be same, noise different

for similar neighborhood, find the average, then replace it for all similar R^2 ~~pixel~~ center.

best average option : look for mean square error.

replace the pixel by average.

difference → creates edges.

sobel → ~~xxx~~ used for edge detection

Hyper-spectral images

Remote sensing → camera, scanner (~~satellight~~, satellites, planes drones camera)

Active → no external source. (LiDAR)

Passive → need external illumination. (hyper-spectral sensor)

RGB camera of phone. → with flash → passive
without flash → active.

Non-local means :→ important for finding patterns in data

repetitions → some similarity.

average neighboring → images

always same pixel, but different noises.

$P_1 = \boxed{P_0} +$ noise
↓
always random.

$P_2 = P_0 + n_2$

$P_N = P_0 + n_n$

always pixel will be same, noise different

noise $= N^2$

↳ যত বেশি average, তত বেশি noise same হবে, same noise different

↳ reduce noise by average.

for similar neighborhood, find the average, then replace it for all similar ~~pixels~~ centers. R^2

best average option : look for mean square error

replace the pixel by average.

difference → creates edges.

sobel → ~~two~~ used for edge detection

<u>Hyper-spectral images</u>

Remote sensing → camera, scanner (~~satellight~~, satellites, planes drones camera)

Active → no external source. (LiDAR)

Passive → need external illumination. (hyper-spectral sensor)

RGB camera of phone. → with flash → passive
without flash → active.

start 2

end 1

10 17 22

31

i can see the the red rose

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

optimal threshold value

space-cnt = start2 - end1 = 22 - 17 = 5

LAB-4 (Compulsory)

within class variance → min. / lowest

~~betw~~ between class variance → MAX

we will get T

or, should the image be changed to gray?

$w_b = \frac{\text{weights for each bar of hist in use}}{\text{Total bar points.}}$ (row, col)

MAX

t = 1

b = 3

10

t = 2

10

t = 3

b = 1

image[image > th] = 255

image[image != 255] = 0

for 0 255

w_b

μ_b

w_f

w_f

max,

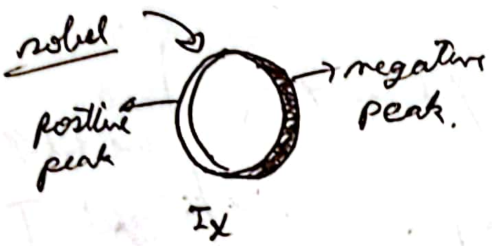
0 to 255 T =

L-4 (Edges & Contour)

Edge detecting Filters:

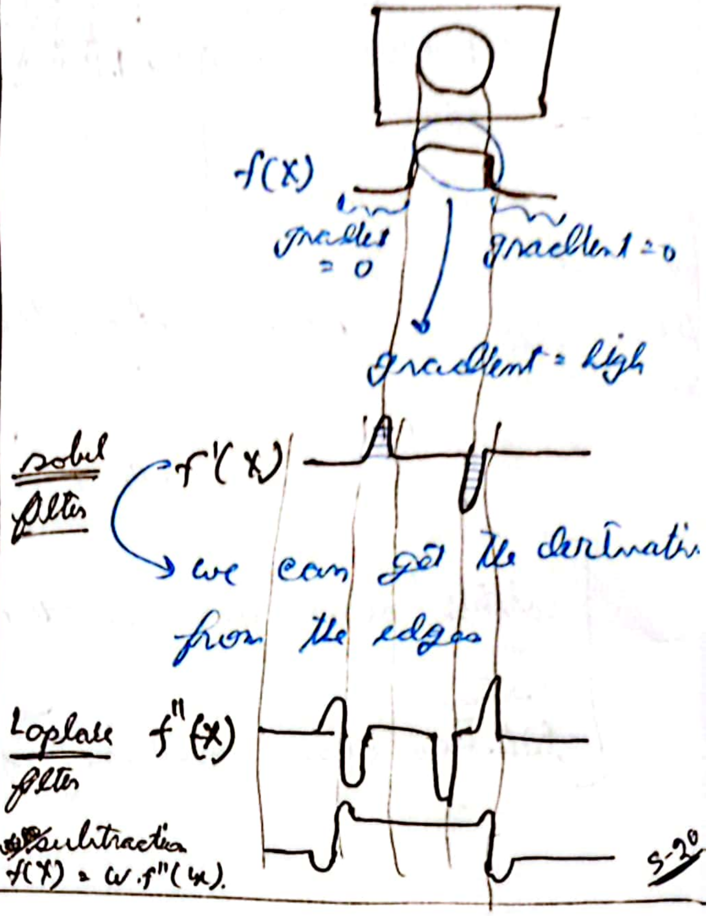
checking derivatives in both the directions.

magnitude of derivative will give the strength of the edges



$H_x = [-1 \ 0 \ 1]$ $H_y = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$

intensity distribution



gradient Filters: → gives result same as computing gradient.

prewitt operator → equal weight to all the pixels

$H_x^P = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ $H_y^P = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$

$H_x^P = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} * [-1 \ 0 \ 1]$

Sobel operator: center line more weight.

$H_x^S = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ $H_y^S = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

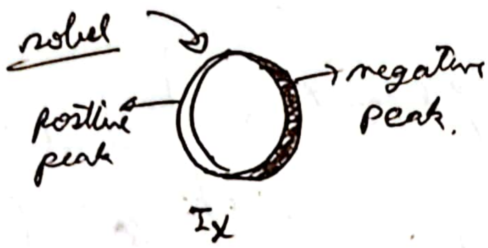
4. 10. 22

L-4 (Edges & Contour)

Edge detecting Filters:

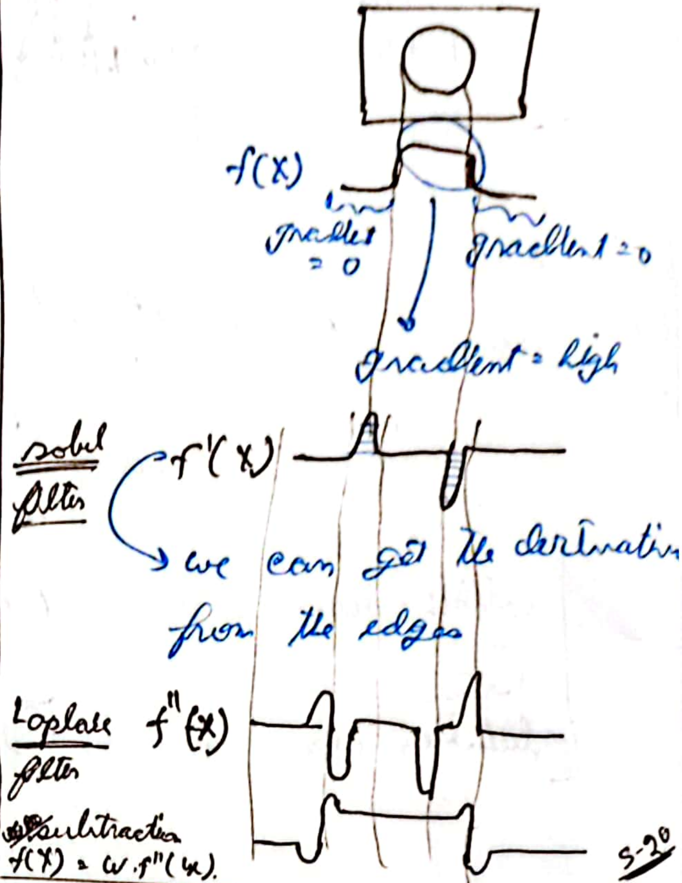
checking derivatives in both the directions.

magnitude of derivative will give the strength of the edges



$H_x = [-1 \ 0 \ 1]$ $H_y = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$

intensity distribution



gradient Filters: → gives result same as computing gradient.

prewitt operator → equal weight to all the pixels

$H_x^P = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ $H_y^P = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$

$H_x^P = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \times [-1 \ 0 \ 1]$

Sobel operator: center line more weight.

$H_x^S = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ $H_y^S = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

· rgb2gray → is similar to taking the mean.

sobel → will show combined x & y (showing edges)

sobel_h → horizontal.

sobel_v → vertical.

· prewitt → also combines horizontal & vertical.

→ cannot do it separately, like sobel.

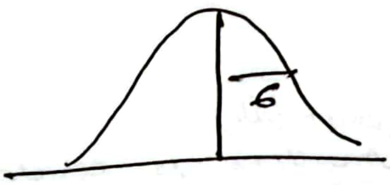
we can compute both magnitude & angles for SOBEL.

Canny filter (popular) → complex & efficient in terms of noises & sharp lines which are not edges.

1st smoothing filter (Gaussian)

then sobel

then thresholding



Gaussian filter.

σ = amount of smoothing

· hysteresis thresholding

→ how the lines are connected.

Read from openCV → tutorial_py_canny.html.

below min_val → not a edge

over max_val → edge.

by default min - 10, max - 25 → 20] not sure check.

---

increase contrast _ sharpness → increasing the intensity of edges.

removing false edges.

max_val] → whether edge or not.
min_val

Gaussian → intensity of blurring / smoothing sigma

sobel

filters are → convolution

$H_x = [1 \; -2 \; 1]$ $H_y = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$

$H^L = H^{laplace} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

$\tilde{f}(x) = \underbrace{f(x)}_{\text{original image}} - \underbrace{\omega}_{\text{weight}} \cdot \underbrace{f''(x)}_{\text{2nd derivative laplace}}$

laplace → pos & neg image $f''(x)$

$\underbrace{\text{astro}}_{f(x)} - \underbrace{1}_{\text{weight}} \times \underbrace{\text{astrolap}}_{f''(X)}$

weight → can be more than 1.

FIJI (image J)

- also
  - make 8 bit (image-type)
- process → filter
  - convolve (edge)

(process → enhance contrast)
- improve contrast
- need to flip histogram.

0 1

enhance contrast
→ equalize histogram.
(stretched histogram)

applying filter is like convolution. → $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

matrix for convolve. Convolution

(mean filter) smoothing is considered as low pass filter.

- mean filter (low pass) → sharp variations will be smothen out.
- only passing low variation

high pass filter → enhance structure. (Only passes sharp variation) edges.
- original image − low-pass-filter.

looks similar to highpass.

image J
- # image − gaussian blur. (with sigma 5) → subtract → original − gaussian (image 1 − image 2) image calculator
- enhance contrast → sharpen. (process → enhance contrast → equality)

filters are → convolution

$H_x = [1 \; -2 \; 1]$ $H_y = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$

$$H^L = H^{Laplace} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$f^*(x) = \underbrace{f(x)}_{\text{original image}} - \underbrace{w}_{\text{weight}} \cdot \underbrace{f''(x)}_{\text{2nd derivative laplace}}$$

laplace → pos & neg image $f''(x)$

astro − [ ] × astro-lap

astro ↓ $f(x)$, [ ] ↓ weight ↓ can be more than 1., astro-lap ↓ $f''(x)$

FIJI (imageJ)

↳ make 8 bit (image-type)

↳ process → filter

↳ convolve (edge)

↳ improve contrast (Process → enhance contrast)

↳ need to flex histogram.

0 1

enhance contrast

↳ equalize histogram.

(stretched histogram)

applying filter is like convolution. → $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

matrix for convolution

(mean filter) smoothing is considered as low pass filter.

↓ mean filter (low pass) → sharp variations will be smoothen out.

↳ only passing low variation

---

high pass filter → enhance structure. (Only passes sharp variation, edges.)

↳ original image − low-pass-filter.

looks similar to highpass.

imageJ

- # image − gaussian blur. (with sigma 5) → subtract → original − gaussian (image1, image2) image calculator
- ↳ enhance contrast → sharpen. (process → enhance contrast → equalize

un-sharpen mask (USM) → similar to high pass filter.
→ for increasing sharpness using edge detection.

$M = I - \bar{I}$

M (ANS) ← original I − smoothed $(I \times H)$ $\bar{I}$ image by kernal

distance for pixels in gray value.

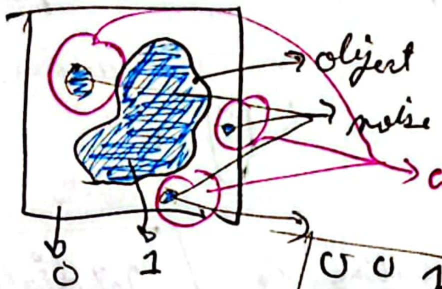
amount = factor a = parameter → weight.

→ sharpen using USM. (a is the sharpening weight)

always define algo ; parameters of your work.

topics
1. smoothing
2. noise reduction
3. edge detection
4. sharpening
5. morphology

Morphology → get rid of noises.



(1) → depending on the threshold, it will remove the noises

$$\begin{matrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{matrix} \rightarrow \text{min-val} = 0$$

1. Apply Min → remove noise
2. Apply MAX → will return object

LAB-5

imread(as_gray = true)

imshow(, cmap='hot')

amount = 2

imread(___ , 'hot')

L5 (Morphology) → binary } both
→ grey } images.
↳ filter use.

11.10.22

change the shape of object.
↳ shrinking & removing noise.

using min filter.
↳ smooth edge, shrink object.

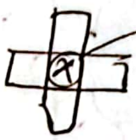
max-filter use করে grow the object.

removing holes. ছবির ভেতরে থাকলে ছোট কালো দাগ দূর করা হয়। এটা দিয়ে ঠিক হয় না।

Erosion (Shrink) – min

anchor point → center of a pixel

Dilation (Grow) – Max filter
↳ bright part বাড়ায়
↳ change anchor pixel.



grow dilation: square → 8 neighborhood.
↓
maximum value selecting. holes are filling in.

1st dilate
↳ will remove holes
→ Then shrink
→ go to original without hole

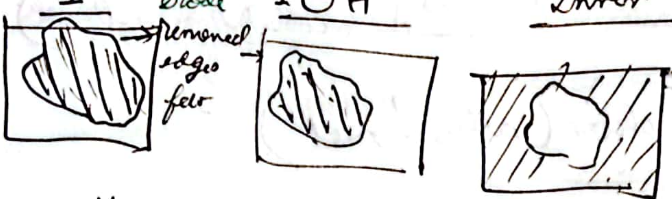
erosion (shrink)

ImageJ 1st edit – invert
process → binary → erode → ~~invert~~
" " " → dilate →

ImageJ Thinks
black → object (foreground)
white → background.

Contour:

I | shrink erode | I ⊖ H | $\overline{I \ominus H}$ Invert

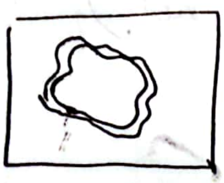


1st shrink will remove holes or noises

Then grow

contour = I $\underline{\vee}$ $\overline{I \ominus H}$

OR মানে union



binary image
0 – background (black)
1 → foreground (white)

dilate – object এর চারপাশে 1 layer pixel add করে (object grow)

shrink → object এর last pixel গুলো identify করে contrast to background. Then remove.

opening → erosion → dilation
closing → dilation → erosion] for cleaning the image.
↳ for reducing holes.

jpeg → lossy compression

processing: ImageJ

8 bit image
↳ select the part → crop
↳ process → adjust → Threshold → otsu
↳ binary → closing (multiple time)
→ invert
→ fill holes (will fill multiple of time until fill) dilate

1-line skeleton of image

looking at background at the object (consider black as object)

If gray image → more weight to center pixels

[13. 10. 22]

L-6

open — close.

T-3

median filter. (thicker)

closing on opened image.

T-2

canny filter → edge detector

invert

canny . 6=1

canny 6=3

invert

L6 # Regions & Objects 18.10.22

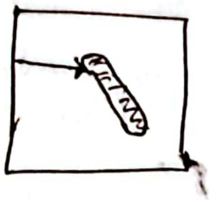
for a ϕ object: pixels should be connected to each other.

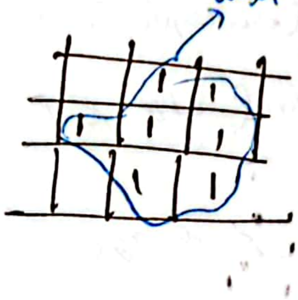
2 methods:

① flood filling → recursive function.
② sequential region marking] old/classical process.

Flood fill:

S-8





will recursively look whether the neighbor is 1 or not. If 1 → add to the set/list.

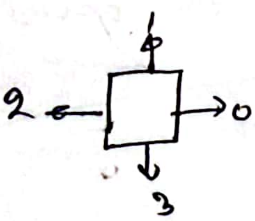
S-9 Inner & outer contour:

→ Run length encoding
  – uses lot of data compression.

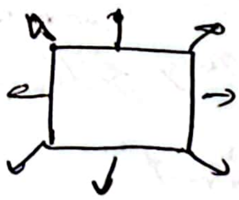
matrix → to RLE (row_ col, length) [compressed way]

S-4 # Chain code:

4-chain code.



8-chain: also has the diagonals



2 objects can have same area but not perimeter.

S-16: Moments: axis changing might be useful for detecting a object.

Real moment

$$M_{ij} = \sum_{x}\sum_{y} X^i Y^j I(X, Y)$$

→ when it is containing the object → all will be 1.

→ binary image এর জন্য শুধু white গোনা দিবে

→ grey image এ intensity দিবে

M_{00} → Area.

Central moment: $\bar{X} = \dfrac{M_{10}}{M_{00} \rightarrow \text{Area}}$, $\bar{Y} = \dfrac{M_{01}}{M_{00}}$

Image J:

sample → blob → [process → binary → make binary.] ✗

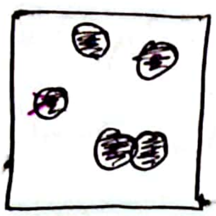
image ✓ → adjust → auto-threshold → otsu

Analyze → Analyze particles → size [50 – infinity] → small noises gone.

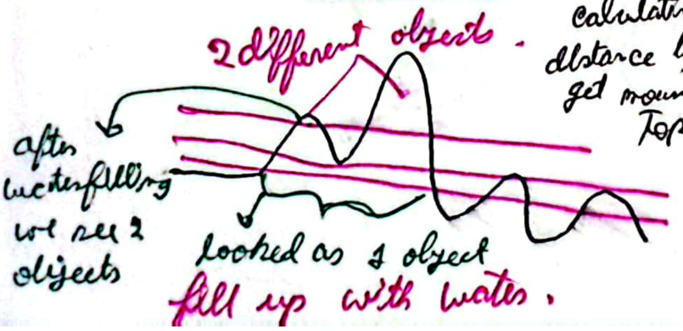
if 0 (small objects will be added)

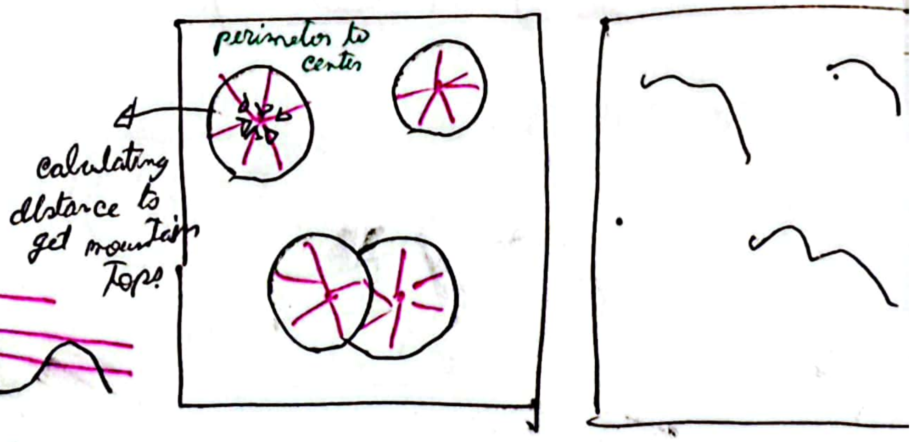
Analyze → set measurement → ① Area, ② center of mass, ③ shape descriptors, ④ perimeter, ⑤ Feret's diameter, ⑥ radius

---



watershed technique





2 objects can have same area but not perimeter.

S-16: **Moments:** axis changing might be useful for detecting a object.

Real moment

$$M_{ij} = \sum_{X} \sum_{Y} X^i Y^j I(X, Y)$$

→ when it is containing the object → all will be 1.

→ binary image-এর ক্ষেত্রে শুধু white হলে দিবে

→ grey image-এ intensity দিবে

$M_{00} \rightarrow$ Area.

Central moment: $\bar{X} = \dfrac{M_{10}}{M_{00} \rightarrow \text{Area}}$, $\bar{Y} = \dfrac{M_{01}}{M_{00}}$

ImageJ:

sample → blob → [process → binary → make binary.] ✗

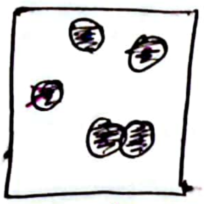
image → adjust → auto-threshold → otsu

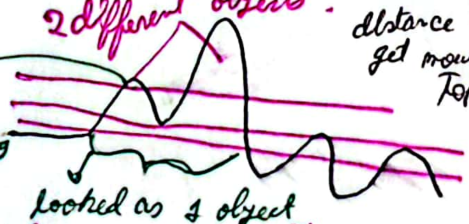
Analyze → Analyze particles → size [50 – infinity] (small noises gone)
→ if 0 (small objects will be added)

Analyze → set measurement → ① Area, ② center of mass, ③ shape descriptors
④ perimeter, ⑤ Feret's diameter, ⑥ label

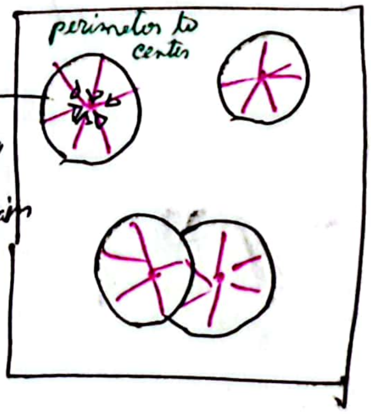
---

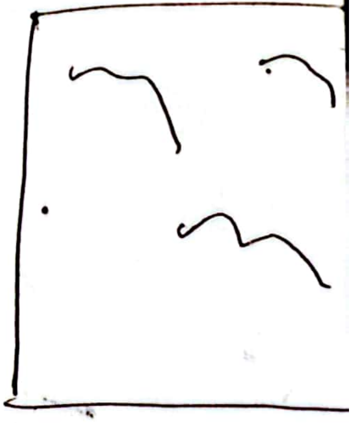
watershed technique





calculating distance to get mountain tops





imageJ!

blob → binary → make binary

→ distant map

distance map — Plugins Analyze → 3D surface plot.

last one is the water levels.

binary → watershed (On the binary image,

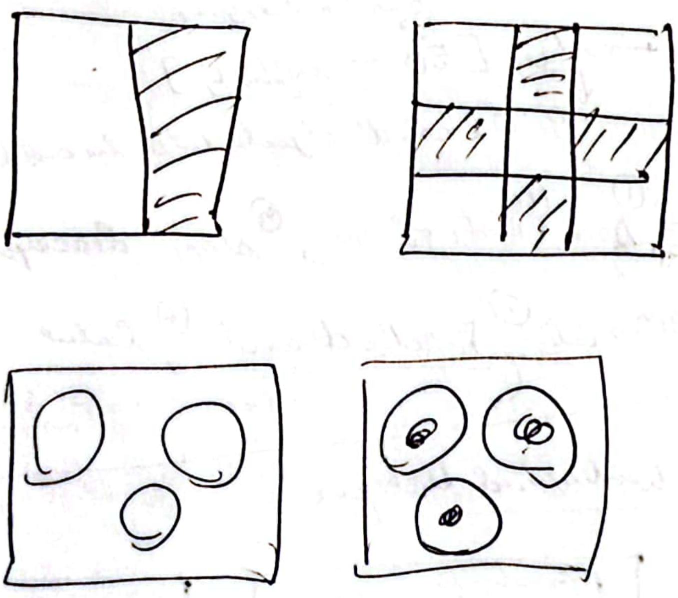
Now it is in segmentation package.

S-32

make a mask

binary image. (object → black
background → 0)

mask = binary + original (background gone)



different ~~str.~~ texture but same area

LAB

full   clr → 0.7   round → 0.9   non-stop

half

X   Y
1279   1171
~~1677~~   ~~1671~~
1583   1444

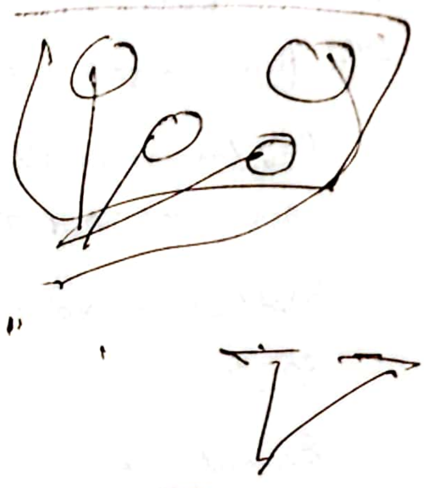
non-stop → round
m8m → longer.

left   (X = 648   1104)   row   col

~~col~~ X   ~~Y~~ row 408 -

6400   4900
↓   ↓
5000   3900

2



1   Malathos
use maximum → filter
making rounder

27 — days.

Area → center of mass

decimal places → 3 to 5.

Circularity will be → 0 to 1 → most round

1st   [0.85   1]   → non stop

2nd → 0   [0, 0.85]   m8m

LAB

25.10.22

L7 Color Images

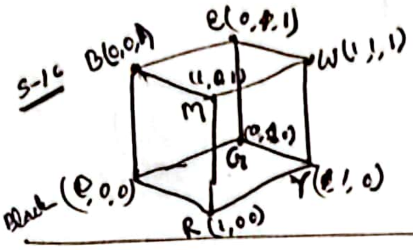
→ fixed for R, G, B
CIE → international commission on Illumination

CIE → horseshoe → colors on edge are unique colors.

eye sensitivity → Red → green → blue
(H) (L)

Compression এর time green & blue কমাই তাহলে Don't have huge effect on eyes

RGB



S-10 **Hue:** → কোন color-এ কিরূপ তাই gray হবে।

Brightness → emitted light per area. যেমন apple-এর সব part-এ আলো সমান লাগে না।

Lightness → value or tone of color.

Chroma → measure of purity of a color.

Gamut → complete subset of colors within a color space.
→ All colors available within a color-space.

S-12 RGB: **Additive colors**: mixture of primary RGB → used in monitors
(→ primary)

CMY: **subtractive colors**: mixture of CMY cyan, magenta, yellow → used in printer devices.
(→ secondary colors)

RGB model

S-18 # **Munsell Color System**: → color modeling এর জন্য use হয়।
– height → intensity.
circle → hue.
Radius → chroma. যত radius হতে দূরে তত বেশি pure (purity)

S-20 **CIE LAB** color model: similar to munsell. They use delta distance - much easier to quantify.

B or → intensity.
axis → a (−G, R→+)
b (+Y, −B)

25.10.23

L7 Color Images

fixed for R, G, B
CIE → international ~~comparator~~ commission on Illumination

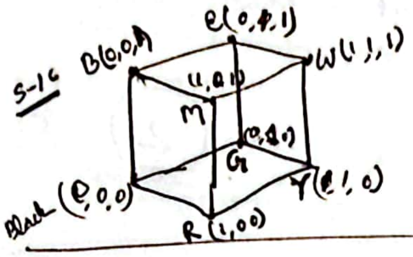
CIE → horseshoe → colors on edge are unique colors.

eye sensitivity → Red → green → blue
(M) (L)

Compression এর time green & blue কমায় কারণ don't have huge effect on eyes

R G B

S-16



S-10. **Hue:** → কোনো color-এ কিছুটা তাই grey হয়।

Brightness → emitted light per area. যেমন apple-এর ওপর part-এ আলো কম বেশি পড়ে।

Lightness → value or tone of color.

Chroma → measure of purity of a color.

Gamut → complete subset of colors within a color space.
→ All colors available within a color-space.

S-12 RGB | **Additive colors** (primary): mixture of primary RGB → used in monitors
CMY | **subtractive color:** (secondary color) mixture of CMY cyan, magenta, yellow → used in printer devices.

RGB model

S-18 # **Munsell Color System:** → color modeling এর কাজে use হয়।
- barn → intensity.
circle → hue.
radius → chroma. যত radius বড় হবে তত বেশি pure (purity)

S-20 **CIE LAB** color model: similar to munsell. They use delta distance - much easier to quantify.

Bar → intensity.
axis → a (−G, R→+)
b (+Y, −B)

ImageJ : (masked out image)

specular reflection এর জন্য holes হয়।

select → crop

image → duplicate.

image → type → 8 → bit

enhance contrast → hist equalization

→ make binary

process → fill holes.

0 → background
256 → object.

Invert → for making mask

(256 → background, 0 → Object)

Img cal → Add → original + mask

Img cam works with wavelength not colors

~~split channel~~

image → color → split channel.

In Red image → Red → whites ~~white~~, green black → black.

yellow → most light.

Convert to LAB

image → color → RGB to CIELAB (এটা just grey করে দিবে। 3D তে করে)

image → type → LAB stack

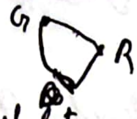
image → color → split channel. (white - L

Green - A (red & green)

red bright, green → black

কালো → B (Blue কালো, yellow → white)

5-25 Gamut : triangle is called color gamut → mix & you will get your color. ~~stu~~ inside the Gamut → সবচেয়ে possible color কোনো edge থাকলে Gamut-এর ২-edge point এর mixture দিবে।



chromaticity diagram-এর 3 corner -এ 3টা color

Triangle-এর বাইরের natural color monitor এ দেখানো যায় না।

ImageJ: (masked out image)

specular reflection এর জন্য holes হয়।

select → crop

image → duplicate.

image → type → 8 → bit

enhance contrast → hist equalization

→ make binary

process → fill holes.

0 → background
256 → object.

Invert → for making mask

(~~0~~ 256 → background, 0 → object)

ImageJ works with wavelength not colors

Img cal → Add → original + mask

split channel

image → color → split channel.

In Red image → Red → ~~white~~ whitish, green/black → black.

yellow → most light.

Convert to LAB

image → color → RGB to CIELAB (এটা just grey হয়ে যাবে। ab এটা হয়)

image → type → LAB stack

image → color → split channel. (white - L

Green - A (Red & Green)

Red bright, green → black.

নীল → B (Blue কালো, yellow → white)

5-25 Gamut : triangle is called color gamut → min & you will get your color. ~~str~~ inside the gamut → তাছাড়া possible color কোনো edge থাকলে gamut-এর ~~ed~~ 2-edge point এর mixture দিবে।

chromaticity diagram -এর 3 corner -এ 3টা color

Triangle-এর বাইরের natural color monitor-এ দেখানো যায় না।

18 Texture Analysis (patterns) [25.10.22]

Texture Analysis

database of different textures.

intensity variations give us texture.

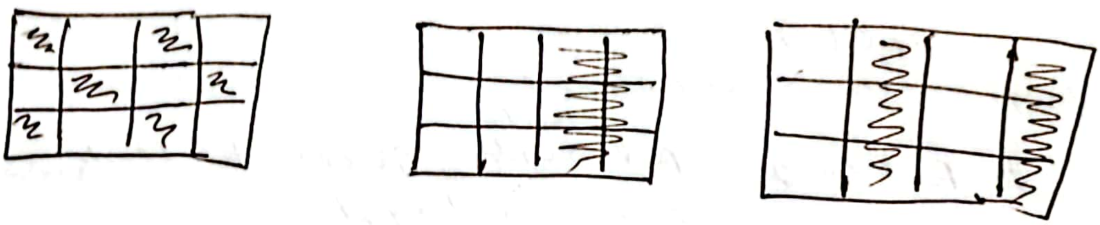
texture analysis → textures on different part of body.

histogram gives info about texture.

Scale → is important. ছোট / বড়ো তে image → তার উপর depend করে complete different texture

texture feature extraction: 2 methods

method 1: ① using histogram (1st order statistics) } → same color.



Prob: in histogram → all 3 patterns will be same, but they are not.

Solution:
histogram of small parts of the images.

method 2 # ② GLCM: looks at neighboring pixels at small levels.

looks at co-occurance of different neighbor pixels.

→ looks at according to angle. (0° → side-side, 45° → diagonal)

gives values for each texture. make a image with the texture values.

(Color & texture) to distinguish between stuff.
Both.

threshold according to gray level co-occurence.

texture & color both important for distinguishing objects

Assignment-02

footprint → row → change the filter.

roughness / bumpiness refers to variations in intensity/ gray levels.

texture analysis: useful when objects are characterized by their texture, compared to intensity.

Traditional thresholding does not work for this.

LDA

① Crop → 8 bit → process → enhance contrast (hist eqa) → binary → fill holes.

import in Fiji

② invert (object → black, back → white)

0 → in object (black)

255 → background

③ add original + mask (image cal) → only the objects

④ RGB image to split: image → color → split channel.

× blue (all black) → red (green-black = black, yellow → white), green

RGB to LAB: RGB to CIELAB (1 gray scale image) ×

↳ cannot split this

✓ image-type (LAB stack) → split channel (works)

× image → color → RGB to CIELAB (B → ANS)

cannot split this

LAB (1 gray image)

---

image → transform

image → LAB stack → take B → image → adjust → threshold → black & white

↓

highlight the object

texture & color both important for distinguishing objects

Assignment-02

footprint → zoom → change the filter.

roughness / bumpiness refers to variations in intensity / gray levels.

texture analysis: useful when objects are characterize by their texture, compared to intensity.

Traditional thresholding does not work for this.

LBA

import in Fiji

0 → in object (black)

255 → background

① (crop → 8 bit → process → enhance contrast (hist eq) → binary → fill holes.

② invert (object → black, back → white)

③ add original + mask (image cal)

→ only the objects

④ RGB image to split : image → color → split channel.

× blue (all black) → red (green-black = black, yellow → white), green

RGB to LAB : RGB to CIELAB (1 gray scale image) ×

↳ cannot split this

image → type (LAB stack) → split channel (works)

× image → color → RGB to CIELAB (B → ANS)

cannot split this

LAB (1 gray image)

---

image → transform

image → LAB stack → take B → image → adjust → threshold → black & white

→ highlight the object

1.11.22

L9 GLCM Texture Features

Histogram → says info about dark & grey.

different textures can have same histogram.

histogram same but texture different.

code from skimage

img_as_ubyte() → float to 8 bit image.

Multivariate Analysis

chemo-metrics → look at data in observative way.
find patterns in the data.

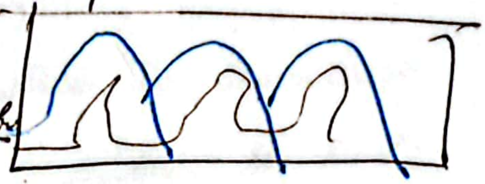
Hoggorm → he wrote most used PCA algorithms
PCA can be programmed in different ways.
pipals is one of the ways

PCA → is the way of finding data with large variation

L10 **Spectral imaging** 8.11.22

small spectral variations cannot be captured in a image.

(green plants reflect in Near-Inf



→ so without the filter → we will not see the original color.

lot of info about the materials in Infra

S-5

Multi-spectral imaging: more than 3, up to 10 discrete bands in a image. ~~pe~~.

hyper-spectral analysis → for finding rotten veg.

hyper-spectral images → usually have more than 3.

λ = wave length.

NEO → gives best high spectral camera

specim, ⇒ swedish company.

line-scan camera:

~~active sensor~~: the camera is → passive.

but using it is active.

$$NDVI = \frac{NIR - R}{NIR + R}$$

-1 to 1

(-1 to 0) → NOT vege

(0.7 - 1) → clearly vegetation.

13.7.23

Phenotyping:

plant breeding → increase production.
→ making crops adaptable to environment.
→ geonme testing for finding the best.
↳ cross geonotyping

Visual selection → sensors
→ genetic markers.

Active sensors → Greenseeker
→ SPAD

passive sensors → dark reference
→ white reference.
→ cameras

① XRAY
low energy → longer wave length. (used for checking the roots)

② visible ↳ blue → shorest wave length.

③ IR → After red → not visible by eyes → uses spectrocopy.

④ Thermal → uses different cameras.

Spectroscopy: → measures different pigments.
→ used for detecting disease.

High-spectral imaging: → each pixel contain multiple info.
→ no of wavelength usually less.

Thermal camera → (has cool) & (un-cool type) → not that expensive
↳ expensive, more accurate ↳ more accurate.

plant under stress → roots look cooler.

3D imaging:

structure from motion (SFM): used in UAV imaging.

TOF (time of flight) & Light Detection & Ranging (LIDAR)
↳ measures time difference for making ptae classification (tree, plant

UAV imaging:

phenomobiles → carts with sensors.

unmanned arial vehicles → drones.

manned arieal vehicles → small planes.

the higher the resolution, lower the image quality.

satellight has low resolution images. 10m per pixel.

drones " high " "

Reason for using drones. UAVS

① independent data acquisition.

multi-rotor, fixed wings are used in drones.

↳ yield estimation.

↳ weed detection

↳ disease detection.

↳ ~~dte~~ biomass estimation.

↳ plant height measurement.

UAV based sensors at NMBU:

multi-spectral

↳ micasense Red Edge - M (Best) [old]

↳ phantom 4 Multi-spectral (easy & accurate to handle) [new]

Photogrammetry:

from 3D to 2D

from 2D to 3D
uses photogrammetry for doing

Auto-mosaic is always not that good.

Examples from field phenotyping at NMBU. (BioSense, Robot image)

⇒ NDVI maps based on drone images.

⇒ Deep learning for image analysis.

CamScanner

Remote sensing

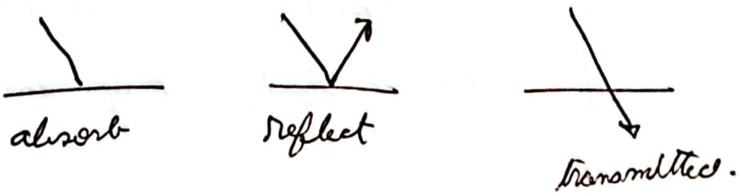
Active → no need of external light source.
no shadow in data.
LIDER (monochromatic) → Light detection & Ranging
↳ measures laser beams when it hits a target.

Passive → need light source. multi / hyper-spectral sensor

→ need sun. (high-spectral image)

Camera → passive → need daylight.

flash → to camera's active sensor



0.4–0.6: clorophill.
1–2.4: → water.

white has more reflectance than black.

5-8 : multi-spectral; 8 less wave lengths. <800 . have sharp edges.
(less bands.)

hyper-spectral: nearly continuous info. 400 – 2000 → wave

more smoother, more bands.

5/4:

color important for classification.

high-spectral → analyze textures.

texture - gray image → calculate M.2.1

Fusion of hyper-spectral & LiDar:

shadow is always detected as PCA 1.

$$NDVI = \frac{NIR - RED}{NIR + RED} \longrightarrow \text{will detect the vegetation.}$$

Multi spectral	Hyper-spectral
→ few bands more than 1	bands follow each other continuously
→ discrete bands	→ full spectrum in each pixel
	→ spectral footprint of each pixel.

[illegible]

Last class

22.11.2022

2020

Next week no Ingun

Task 1: python & fiji both fine.

explain what you are doing.

You can also choose Inplace

Write code.

if you are unable to do → write pseudo code

USM Ismooth → I * gauss

$I_{usm} = I + a(I - I_{smooth})$

Image J

↳ convert to 8 bit

↳ Filters → unsharp mask

→ choose sigma.

$\alpha = 7$

Write write observations with different values used.

Task 3: Thermal camera used to detect fault in solar panels.

Image J

1. 8 bit
2. split chanel
3. blue → easier to threshold (Best)

Image — LAB stack

↳ none good enough

Illustrate what kind of operations you can do.

-green + red (might work

5f # keep binlis one
" " .] for both images.

You are allowed to crop image
↳ for removing unnecessary parts

make the histogram & say its not a good idea.

= local otsu → thresholds a lot of noise
= global otsu → works better.

Task 3 UV flurosence

In dark, use a UV torch to take photos.

fluroscents present in the panels will react with the UV torch and show.

Oxygen will come out of the cracks and destroy fluroscents → to find cracks.

Image 7:

process 1 color - split channel → blue (best). → not that different

process 2 - image → type - lab stack - LAB (L image 1st is best)
↳ stack to images → take L image.
↳ make 8 bit → or
↳ enhance contrast.

5)

\# Keep kibo one
\# " " .] for both images.

You are allowed to crop image
↳ for removing unnecessary parts

make the histogram & say its not a good idea.

= local otsu → thresholds a lot of noise
= global otsu → works better.

Task 3 UV flurosence

In dark, use a UV torch to take photos.

fluroscents present in the panels will react with the UV torch and show.

Oxygen will come out of the cracks and destroy fluroscents → to find cracks.

Image 7:

process color – split channel → blue (best). → not that differ

process 2 : image → type – lab stack – LAB (L image 1st is best)
↳ stack to images → take L image.
↳ make 8 bit →
↳ enhance contrast.

prous 1: take blue. (Better than 2)

image → auto-threshold → try all

→ try Otsu (cracks are seen)

black parts → dark on UV lights.

EXAM start Monday 5th
deadline Sunday 11 th

→ Describe what you try

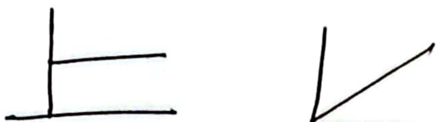
→ what you have done

→ write your understanding

Say you have did exploration in image 1 and chose this image and did rest of the code on the selected image.

→ histogram

→ then cumulative hist. (not ideal equalized histogram)



hist   cumu hist

→ after stretching → again → histogram.

when thresholding → both local and global threshold.

↑ local → not a good idea. thresholds all the noisy part.

— very low signal

→ will amplify the noises.

— global — ✓

cracks ([illegible])

Task-4

load numpy file →

[you can crop unnecessary part
↳ if it is too big for your PC.

RGB image

~~R G B~~ → order for the image

In class lecture wrong.

[blue (400 → 650) peak 450
green → peak 550
red → peak 650
↳ take the corresponding band → in file it says from band 1.

[water is black
solar panel blue
vege → green & yellow

Better than 1 band.

= more accurate compare to 1 band.

more bands to average

[Range of bands 10:20 — mean along axis ?
40:50 —

1 band →

a pixel

spectrum image → explain the image

asplat]
grass } flat → because dark all over the area

put legends and axis MUST
title #

→ take mean along X and Y axis. (axis = 0)

20 pixel

mean spectrum : → area of 20 × 20
↳ mean spectrum

→ it will make the graph much smoother (spectrum)

easier to see and understand

NDVI → manual & function both allowed,
she did both.

vmin, vmax → adjust for better clearly.
0 0.7 (১-৫ দিলে ভাল হয়)

সবগুলোর হবে। দিলে better comparison. এ ভাবে হয়

greener → the better.
dark green → lot of clorophill.
কিছু জায়গা → not healthy → less green.
↳ misclassification. ও হতে পারে।

(5) PCA score

score images → PCA images

↳ check if there is at least 3 component.

0, 1, 2 → 3 টা

0.99 → 55 PC এসে

0.999 → 161 PC এসে

0.95 → 1st তিন নিয়ে এসে

↓ or lower.

adjust vmin & vmax

↳

PC3 → solar panels and some vegetation.

PC2 → only vegetation, no solar panel.

Can look at PC3 or PC4 → for understanding.

PC3 → very noisy (not much information)

loadings

loading 2 PC2 → (solar panel) pair

↳ for score image 2

wavelength are variables

pixels of PC-score sample.

solar panels very high value. so might be in the range of (0-75) or (____).

Task 8 K-means → from PC component OR
from whole image. (might be too noisy)

→ might have better result on reduced

can do both and see differences.

does not work good, not optimal.

→ check → loadings from K means.

उसका graph देख।

Task 7: # Do gaussian classification.

Blue roof

~~337~~ ~~419~~
342 433
211 7
214 12

ground truths.

green → road solar panel.
red → vegetation.
pink → dark road in front of solar panel.

dark tree

247 310
272 329

254 312
271 317

155 230
162 239

grass

280 258
293 282

asphalt

80 148
90 169

dark roof

~~97~~ ~~284~~ 92 29
~~106~~ ~~303~~ 103 30

orange roof

~~154~~ ~~226~~
~~166~~ 233

dark tree

253 313
269 318 ~~326~~

254 310
265 318

roof brown

160 232
166 239

307 129
315 133

dark road

271 190
282 206

290 275
310 295

85 150
105 170

90 300
110 320

low signal image এর contrast বাড়ালে
noise amplify হয়ে যায়।

spectral foot prints / are same → different curves in loading plots.

team : platform

Q. spectra → Y-axis = ?

Q. PCA → percentage = ?.?
just take required PCAs = ?

loading plot = ? explanation = ??

which band no you get a good wavelength.

PCA2 → vegetation

which band contribute to most variance = ?

white object

black back ground

L-1

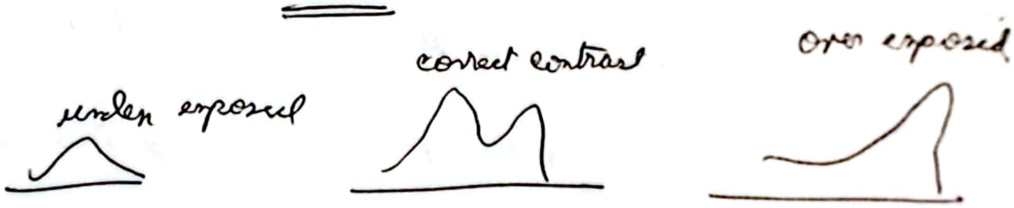
human . can discriminate less → brightness change
more → intensity change

contrast: same intensity → but outer layer এর জন্য different লাগে। contrast helps us identify

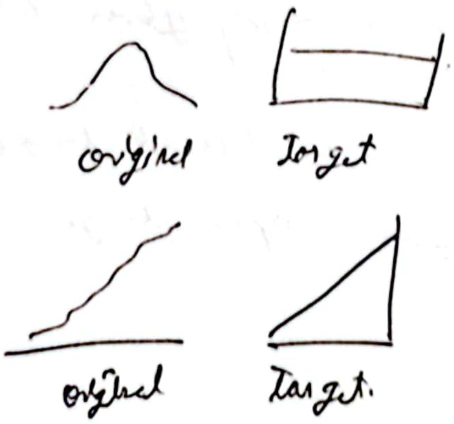
pixel value : gray ↓ 0-255 | RGB 0-255, 0-255, 0-255 (R, G, B)

Humans seperate R-G better than Blue-Yellow
Red-Green

L-2



hist-equalisation → helps with increasing contrast



[brightness +, -
contrast percentage

hist equal ; increase contrast
↳ stretches the contrast
↳ skewness →

hist of binary → 2 bars

① thresholding → as a point operation
② inversion →

L-1

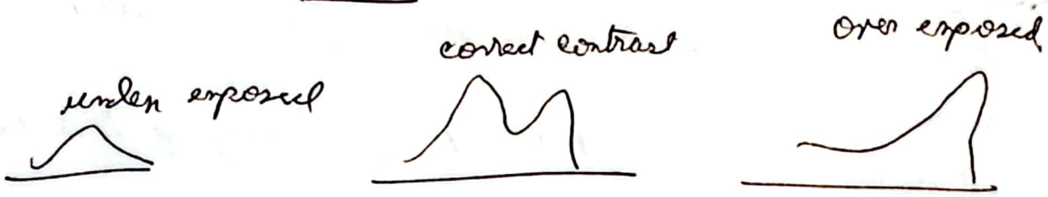
human can discriminate less → brightness change
more → intensity change

Contrast: same intensity → but outer layer এর জন্য different লাগে। contrast helps us identify

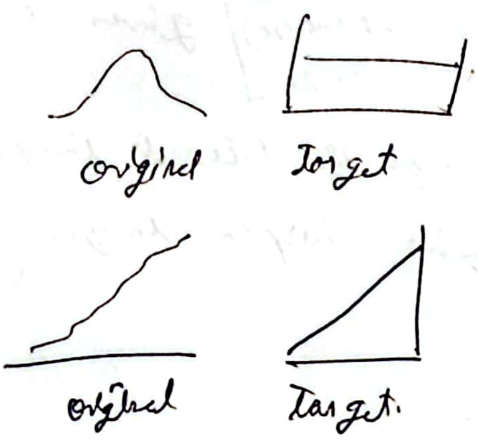
pixel value : gray ↓ 0-255 | RGB 0-255, 0-255, 0-255 (R, G, B)

Humans seperate R-G better than Blue-Yellow
Red-Green

L-2



hist-equalisation → helps with increasing contrast



[brightness +, -
contrast percentage

hist equal ; increase contrast
↳ stretching the contrast
↳ skewness →
hist of binary → 2 bars

① thresholding → is a point operation
② inversion

L-3 (Filter)

Filter / point operation → works on 1 pixel.

works on multiple pixels.

→ to blur & sharpen → need filter

~~lst~~ blurring / smoothing filter:

① mean filter → takes the mean from range.

non-linear ② max filter → takes the max from range. (white वाले)

③ median filter → sorts and takes the mid one.

non-linear ④ minimum filter → takes the min from range (black वाले)

Problem of linear filter → smoothes edges, & lines.

reduces image quality.

⑤ Gaussian: (Best)

Compared to linear → non-linear filter better.

non-local-means → better at preserving of textures

→ denoise_nl_means

just mean → is not local.

convolve = ?

[residual = original − filtered image (denoised image)] gives edges.

local → denoising filter → on the blurred image

gives sharpen image.

lrly

Denoising filters

original - denoised = edges

Best ① Gaussian → sigma ∞ ∞ blur.

3rd ② median [~~pref~~ preserve the edges while smoothing]

③ mean

2nd best ④ non-local.

⑤

median → sharpen = ? removes the noise thats why looks sharper

non-local → uses mean filter

L-4 (edge) → Gaussian ones are best
& Contour

edge → ~~when~~ image positions where the local intensity vary dramatically in relation to its surroundings.

edge
2nd worst
① Prewitt filter → 1 in the center line & col.

$$H_x^P = \begin{bmatrix}1\\1\\1\end{bmatrix} * \begin{bmatrix}-1 & 0 & 1\end{bmatrix} \qquad H_y^P = \begin{bmatrix}1 & 1 & 1\end{bmatrix}\begin{bmatrix}-1\\0\\1\end{bmatrix}$$

edge
2nd best
② Sobel filter → higher weight to center line & column than prewitt.

$$H_x^S = \begin{bmatrix}-1 & 0 & 1\\-2 & 0 & 2\\-1 & 0 & 1\end{bmatrix} \qquad H_y^S = \begin{bmatrix}-1 & -2 & -1\\0 & 0 & 0\\1 & 2 & 1\end{bmatrix}$$

edge
Best
③ Canny filter : (multi stage edge detector) : uses Gaussian.

→ sigma of gaussian smoothing
→ hysterisis thresholding (min & max)

low sigma → edges with lot of noise.
high sigma → just the perfect edge.

L-4 (edge) → Gaussian ones are best & Contour

edge → ~~when~~ image positions where the local intensity vary dramatically in relation to its surroundings.

edge

2nd worst ① **Prewitt filter** → 1 in the center line & col.

$$H_x^P = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \qquad H_y^P = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

edge

2nd best ② **Sobel filter** → higher weight to center line & column than prewitt.

$$H_x^S = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad H_y^S = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

edge

Best ③ **Canny filter**: (multi stage edge detector): uses Gaussian.

- → sigma of gaussian smoothing
- → hysteresis thresholding (min & max)

low sigma → edges with lot of noise

high sigma → just the perfect edge.

worst edge used for sharpening

④ Laplace filter: uses $f''(x) \rightarrow$ 2nd derivative.

X used for sharpening

⑤ Low pass filter (L) passes low frequencies, coarse structure, long wavelengths.

$$L = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

sharpening

2nd best ⑥ High pass filter (H): passes high / fine structures / short wavelengths.

→ Makes image sharp.

$$H = I - L$$

$$= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \frac{1}{9}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

sharpening

uses Gaussian

Best ② Unsharpen Mask (USM): sharpening using edge detectors.

sharped image → original image + smoothed (unsharped) image.

a → weight factor → that controls sharpening.

mask = $M = I - (I * \text{smoothing, unsharpening filter})$

sharpened image, $I = I + a M$ → amount of sharpen

③ Laplace sharpening: does not work that good.

L-5 (morphology) → used on binary image.

→ modify shape

Binary image - (~~Skeletonize~~

1 → pixel in foreground (objects) [white]

0 → background. [black]

~~Median filter~~:

- Kernel → is the area size defined by us.
  - ↳ disk, square, diamond, cube. ~~ball~~. (ball for 3D)
- Range (zone) → for the filter to work.

① Dilation:
- → takes max over the kernel ~~area~~
- : removes holes. (grow)
- → usually square/disk

② Erosion:
- → removes small objects (shrink)
- → makes holes huge
- → takes min over the kernel area.

Going back to old size.

~~Contour~~: Borders/edge of the image.

~~Erosion~~: Shrink - Grow

① Open (Erosion - Dilation):

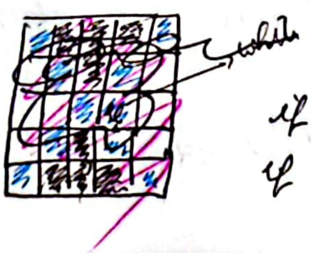
→ smoothes objects, removes noises.

② Close (Dilation - Erosion)
(Grow - Shrink)

→ smooths objects, fills in small holes

Thinning → shrinks the object, unless 1 pixel is left

Skeletonize: → Object should be white, background black (of ~~edge~~ contours)



If object white → skeleton → correct

If object black → skeleton → wrong

Filters → can be used on both gray & binary image.

① max filter → dilation / grow : Process → Filters → Minimum
→ takes max value of the local zone.

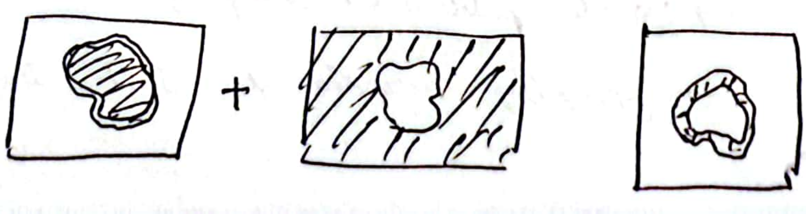
② min filter → Erosion / shrink : Process → Filters → maximum.
→ Takes min values of local zone.

- S-20! Covert to mask?

③ Median filter: preserves edges while removing noise.
↳ Blurs image, but preserves edges.
↳ Kernel zone → ~~sort~~ sort ~~???~~ takes the median in the anchor point.
↳ used for smoothing heavy noise / pixelated image.

Contours: object black → ~~dilate~~ shrink (1 pixel)
→ invert the shrink. (object white small)

Original (black) + inverted small (white) = contours.



RGB → R →
G →
B →

CIE LAB → L →
A →
B →

L-8 (Texture Analysis)

Statistics on pixel value / Texture feature extraction:

Type 1: (1) histogram → mean
max

(1st order statistics)

problem → does not tells us anything
→ different texture, same histogram

solution
→ take patches do small hist
→ or do GLCM

Higher order statistics

Type 2: GLCM: Grey level co-occurance matrix

plot & see → the texture classification.

→ pattern

→ counts patterns or ~~group of~~ re-occuring pixels pairs

→ grouping according to similar texture

L-11 (Spectral Imaging)

(Visible light)
HUMAN Vision: 400 – 700 nm

~~RED: 410 – 490~~
~~510 – 680~~
~~GREEN: 490 – 680~~
~~BLUE: 420 – 490~~

Hyper-spectral camera:

ultra violet: 200 – 400 nm

Visible light: 400 – 700 nm

Near Infrared: 700 – 1300 nm.

Hyper-spectral data: gives a detailed spectrum of channels in the (1) visible and (2) near infrared and (3) shortwave infrared, bringing out the invisible differences in the surface material.

Human Vision / Visible light: 400 – 700 nm

RED: 650 – 70 nm [Aprox: 645 nm]
GRENN: 490 – 575 nm [Approx: 510 nm]
BLUE: 420 – 490 nm [Approx: 440 nm]

Classification: Assigned predefined labels to objects based on their features.

Clustering: no of clusters will be told. Grouped according to their similarity present.

Classification

① **Supervised** → user chose the sample pixels in images.

→ Gaussian maximum likelihood classification (GMLC)

② **Unsupervised** → grouping based on common characteristics without ~~without~~ any sample.

→ K-means.

PCA: in score images : black → background

color → ~~t~~ items / objects visible in that PCA

look at loadings & score & and they there's peak or high here because these things are visible on the score image.

→ vegetation reflect in green.

Green 490 : 575 : → small bump in this point confirms vegetation. # After Red peak to near infrared (vegetation

all other objects : flat spectrum

black → low score

other → high score

specific slope of vegetation : red low for vegetation.

NIR high " "

Classification

① Supervised → user chose the sample pixels in images.

→ Gaussian maximum likelihood classification (GMLC)

② Unsupervised → grouping based on common characteristics without without any sample.

→ K-means.

PCA: in score images: black → background

color → items/objects visible in that PCA

look at loadings & score & and they there's peak or high here because these things are visible on the score image.

→ vegetation reflect in green.

Green 490 : 575 : → small bump in this point confirms vegetation. # After Red peak to near infrared (vegetation)

all other objects: flat spectrum

black → low score

other → high score

specific slope of vegetation : Red low for vegetation.

NIR high " "

~~Spec~~ Score Image

light → high score
dark → low score.   ] opposite from each other in PCA scale.

<u>loadings</u> corresponds to wavelengths. By loadings, we can see which wavelength have been given the weight to generate score image.

<u>PC0</u> → flipped from vegetation. Instead of peak at green, fall at green.

negative value

spectral signature

Cannot say directly which PCA it is.

Y → intensity (arbitrary unit)

<u>PC1</u> → <u>vegetation</u> low at red high at NIR

PC0 → <u>opposite of vegetation.</u> (vegetation is black)

<u>score color image</u>

grass

402 40
422 76

asphalt

253 769
268 791

black roof

719 735
742 770

red roof

133 886
150 894

solar

405 429
411 439

water

713 896
751 944

GIMLE

grass

387 11
432 61

white roof

1st
300 330
341 366

2nd
379 590
395 625

soil ground

961 641
982 702

asphalt (4 way)

252 772
272 794

black
dark roof

258 520
283 553

road [illegible] match
concrete ground

316 232
347 287

tree

122 71
157 117

green roof

296 656
324 685

red roof top

133 863
149 882

127 620
136 636

**

alpha = 40

· blur

nd image

·alpha =

sharpened = blurred
+ alpha * (blurred_f -
filter blurred)

155 439

2732 3643

multi-convolves

Task1: 2|3 → amount

① multi-convolve. (~~contrast~~)
↓
LOG filter

② median filter → keeps the edges okay.

① Laplace
② USM → sigma, ~~tab~~ amount.
③ High

median on RGB → Try

On gray too

Task2: a) should it be on the RGB or the gray image (Imgun)

LAB
↓
A

a) Li → does not have shadow (Gray image)

	it	count.
dilate →	10 →	1
eradicate	8 →	1

Default | 1 | 1 |

mask
black balls
white background

87
7

94

take image

→ crop

→ LAB → choose A (without shadow)

C2

2nd img

threshold → Li

→ lot of holes / dots.

↳ dilate → it (10) → count 1

↳ erode → it (8) → count 1 [rid of holes few remains]

→

closing → 8 iter

2 (C)

fill holes

watershed

Im

V3 → works without shadow