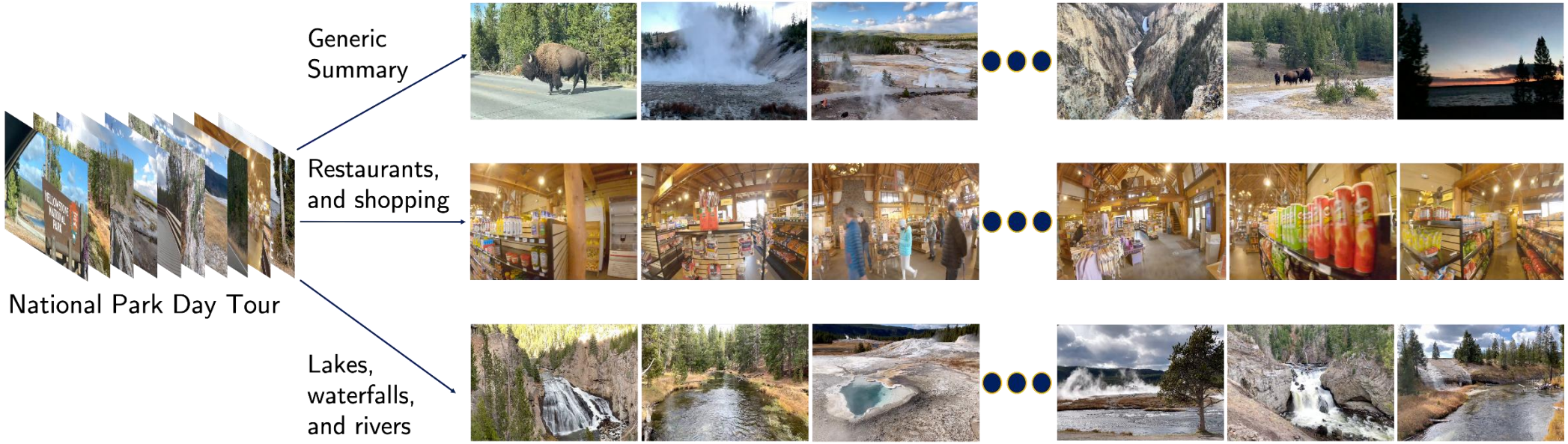
Uncited images created by the author using CoPilot or Adobe FireFly or Flux1.ai

SINGAPORE INSTITUTE OF TECHNOLOGY

INF2009 – Edge Computing and Analytics [2024/25 T2]

Edge Analytics – Image and Video

Image/Video Analytics

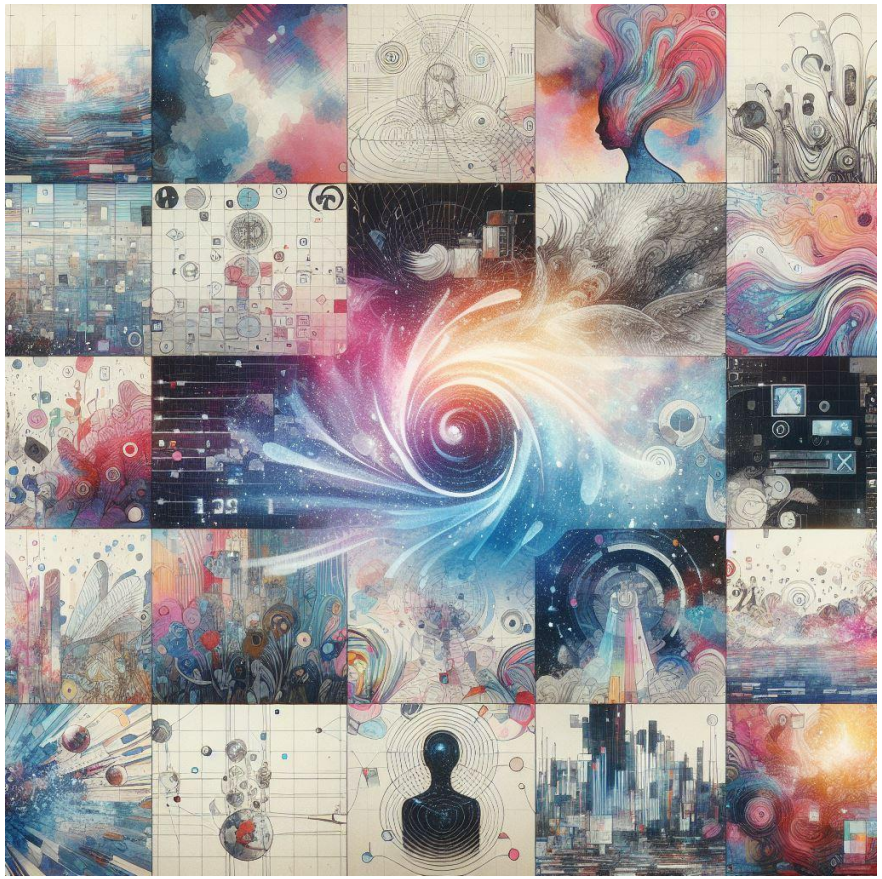


© https://medhini.github.io/clip_it/

Reference:

- **Szeliski, Richard. Computer vision: algorithms and applications. Springer Nature, 2022.**
- **Gonzales, Rafael C., and Paul Wintz. Digital image processing. Addison-Wesley Longman Publishing Co., Inc., 1987.**
- **https://opencv.org/**
- **https://scikit-image.org/**
- **https://developers.google.com/mediapipe/**
- **https://towardsdatascience.com/exploring-ia-at-the-edge-b30a550456db**

Image/Video Analytics on Edge - Applications

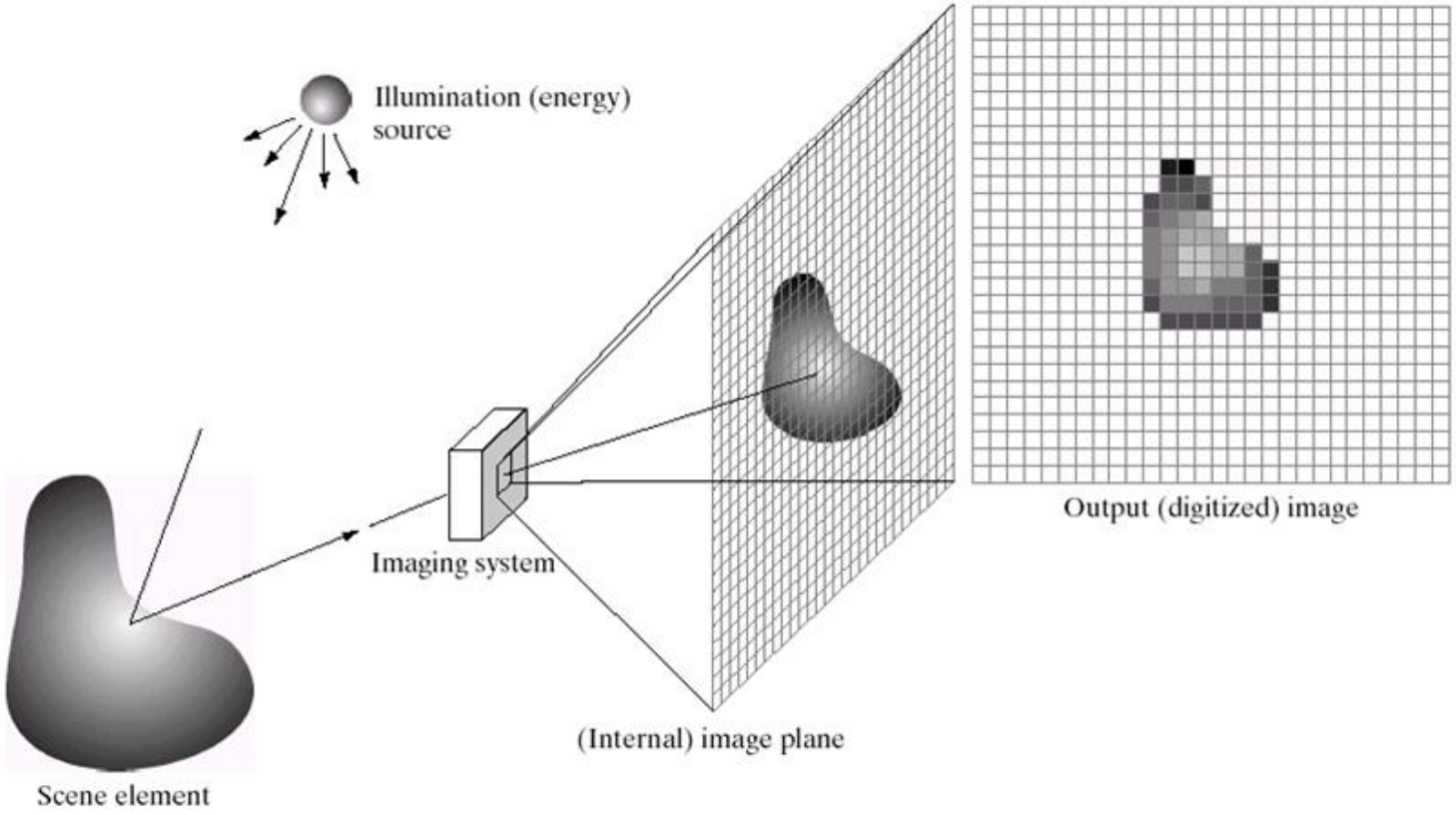


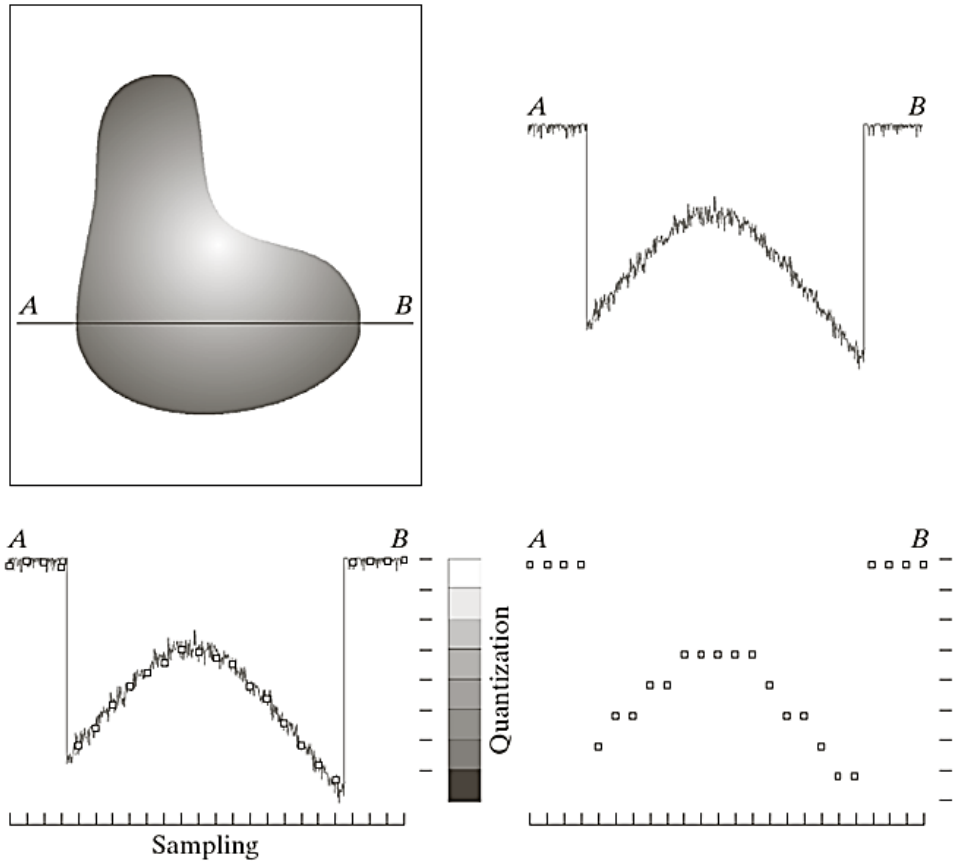
- *Scene Classification*
  - *e.g. Surveillance*
- *Scene Segmentation and Detection*
  - *e.g. Traffic Scene Analysis*
- *Human Computer Interface*
  - *Hand Gesture Recognition*
  - *Facial Landmark Recognition*
- *Video Summarization*
- *Motion/Depth Estimation*

Content

- *Image/Video Acquisition and Parameters*
- *Image Features – Edges, Shapes and Textures*
- *Hand and Facial Landmarks Recognition*
- *Motion Estimation and Applications*
- *Lightweight Machine Learning for Image/Video*

Image Formation – Spatial and Intensity Resolution





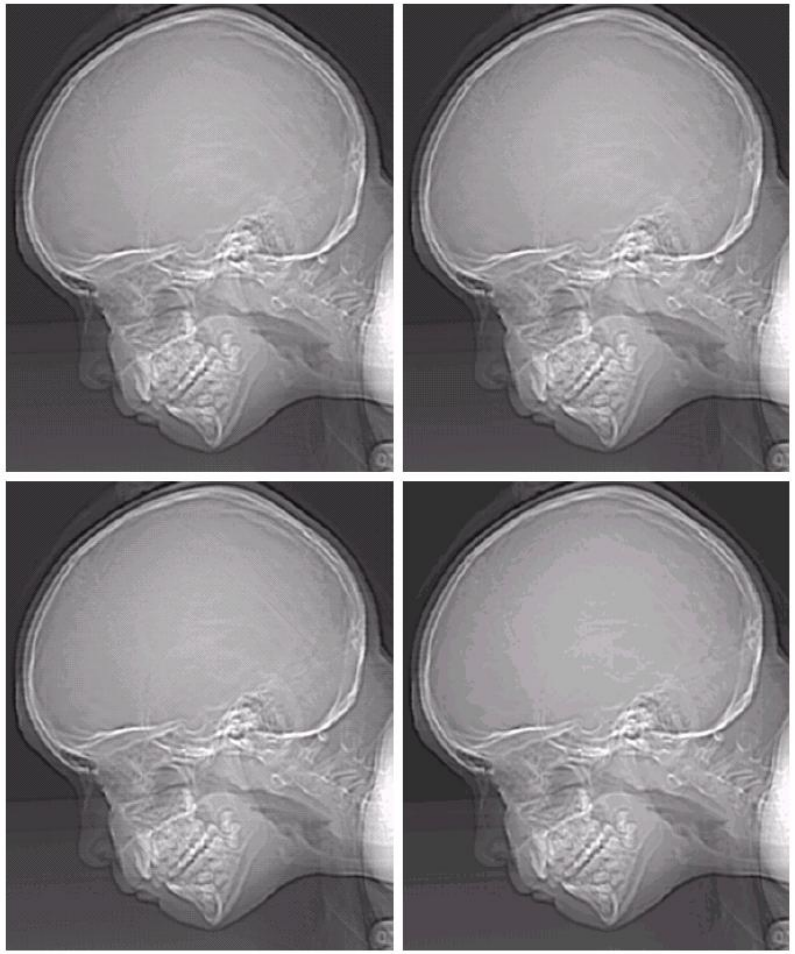
Spatial Resolution → Number of pixels per area
Intensity Resolution → Number of intensity levels (bits)

Image Formation – Spatial and Intensity Resolution

*dpi = dots (pixels) per inch

a b
c d

FIGURE 2.20 Typical effects of reducing spatial resolution. Images shown at: (a) 1250 dpi, (b) 300 dpi, (c) 150 dpi, and (d) 72 dpi. The thin black borders were added for clarity. They are not part of the data.



a b
c d

FIGURE 2.21 (a) 452 × 374, 256-level image. (b)–(d) Image displayed in 128, 64, and 32 gray levels, while keeping the spatial resolution constant.

Histogram → Connecting Pixels & Intensity

Histogram of an image is defined as follows:

$$\boldsymbol{h(D) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(x,y) == D} \qquad \boldsymbol{D \in \{0, 2^L - 1\}}$$ ***for unsigned L-bit image representation***

(# of pixels of I(x,y) that have intensity D)

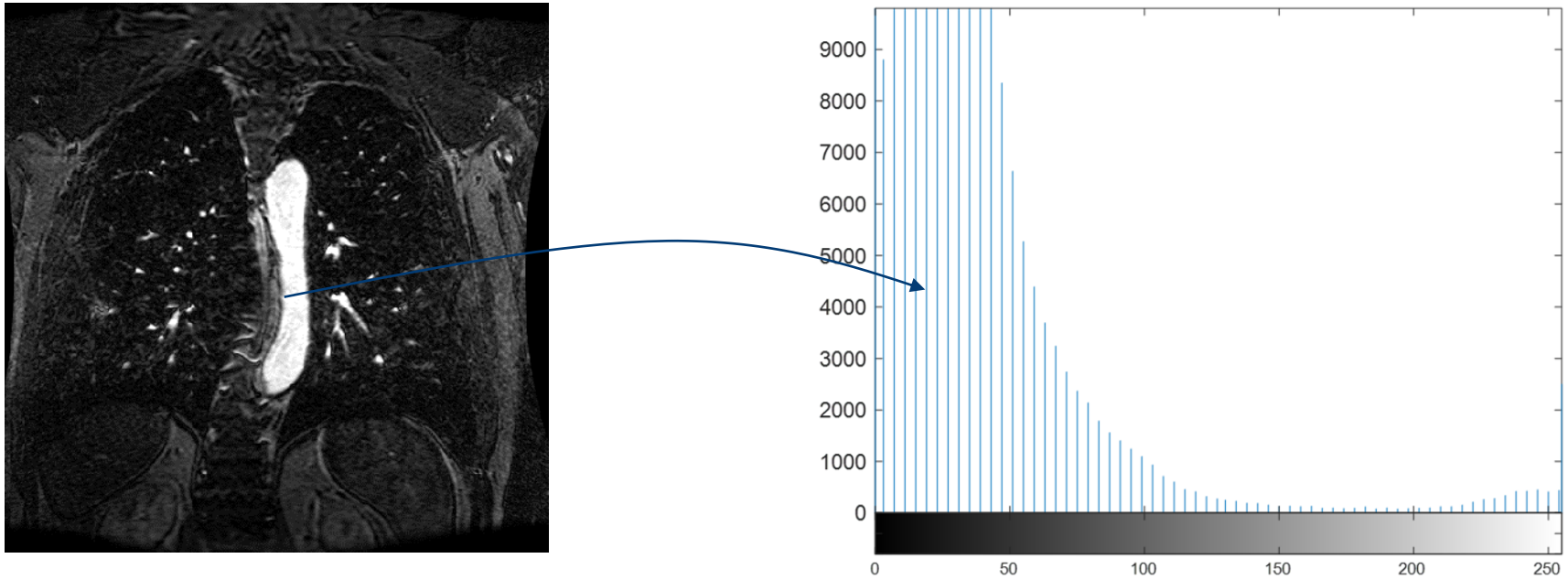
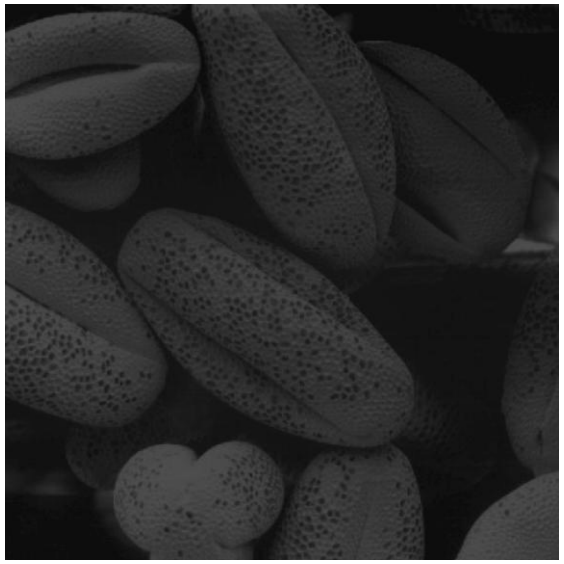
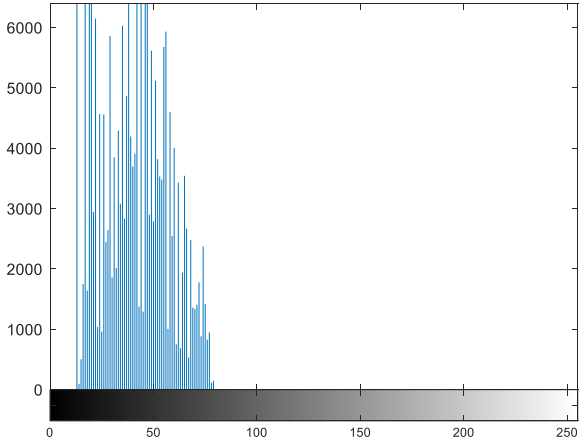


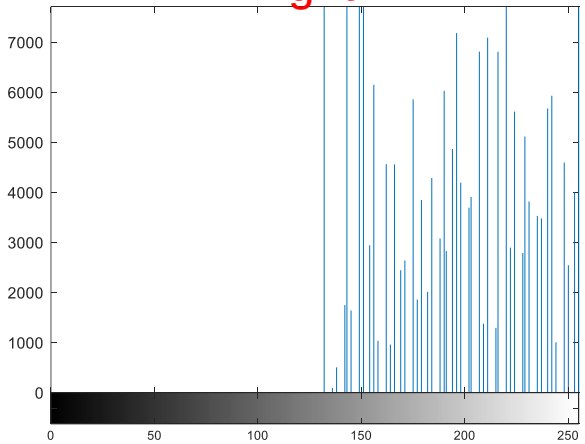
Image Contrast

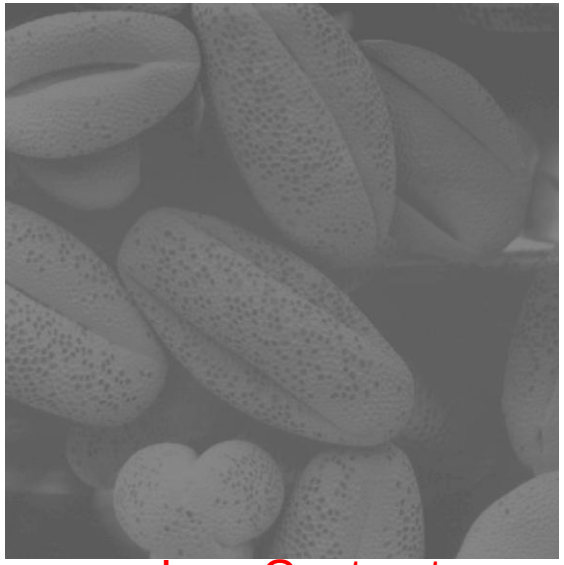


Dark

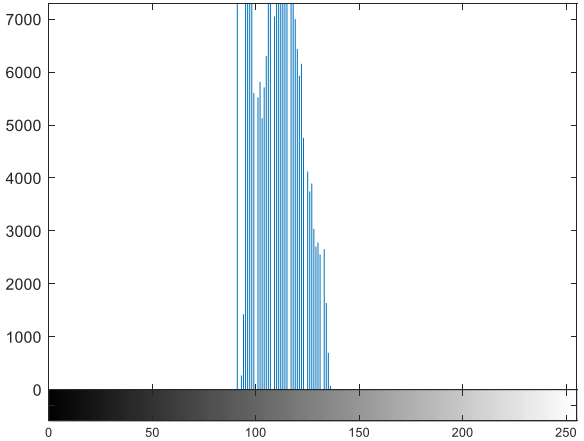


Light





Low Contrast



High Contrast

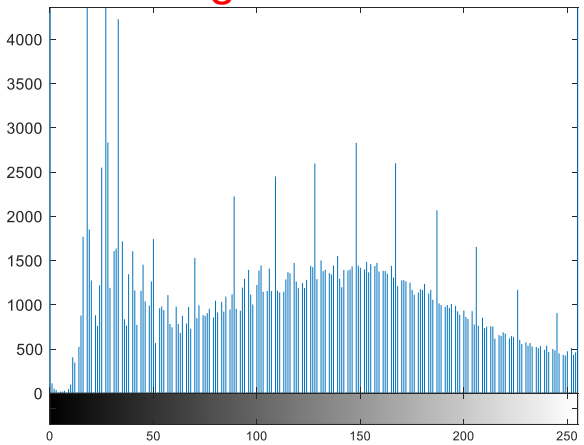
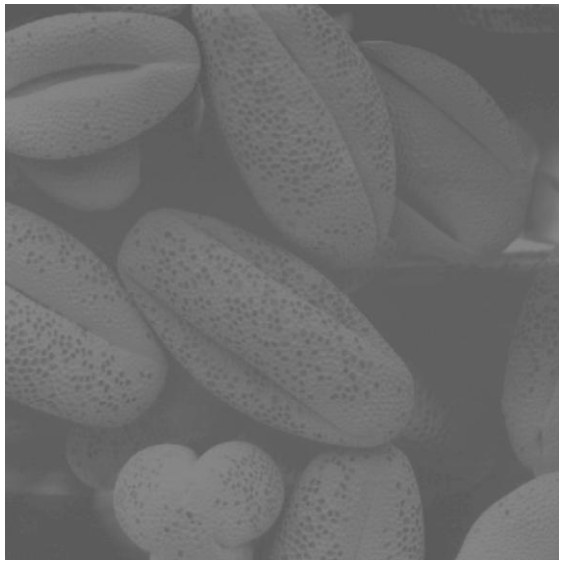
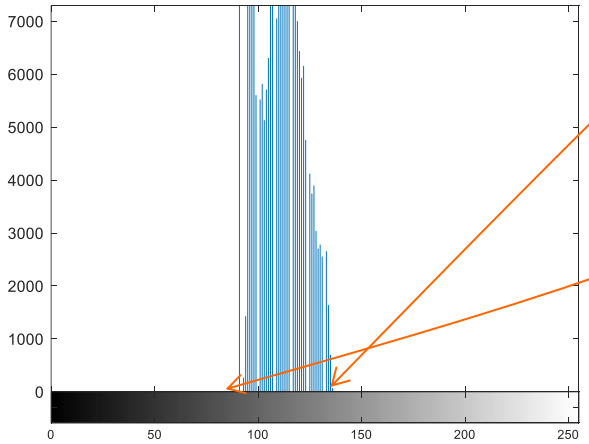


Image Preprocessing – Contrast Stretching

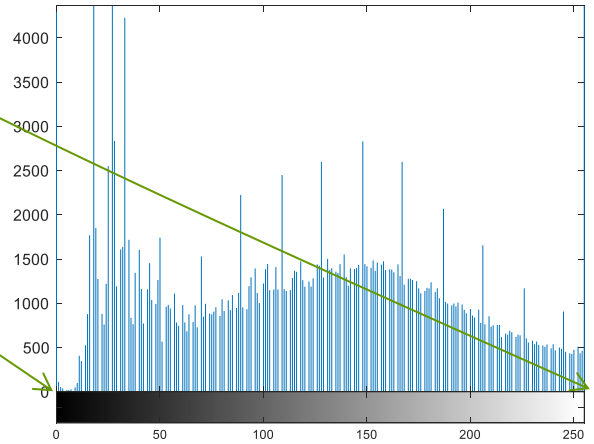


$$J(x,y) = 255\,\frac{(I(x,y) - L)}{H - L}$$

Low Contrast

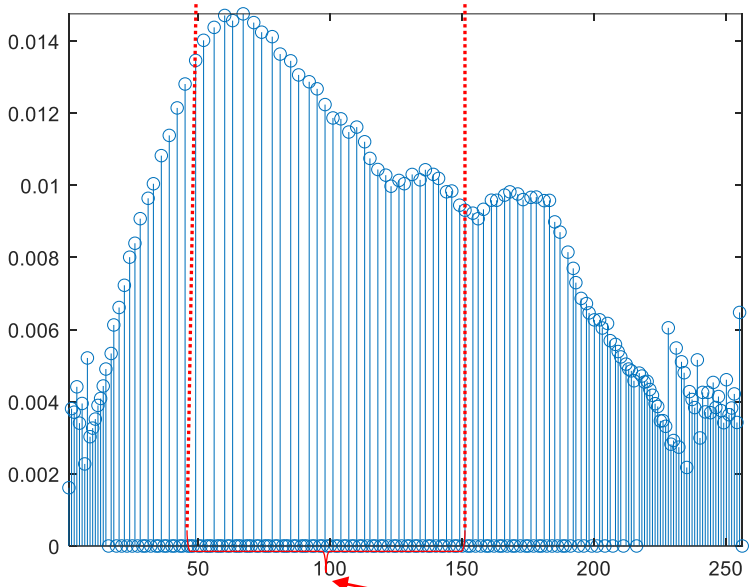


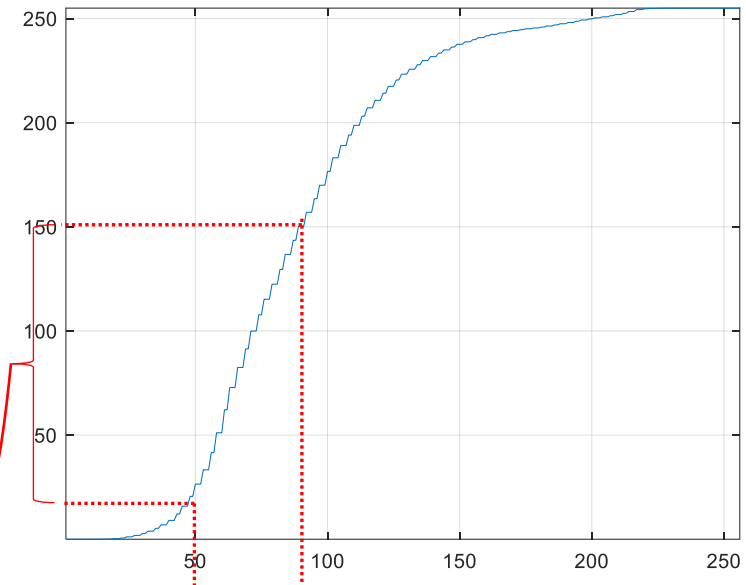
High Contrast



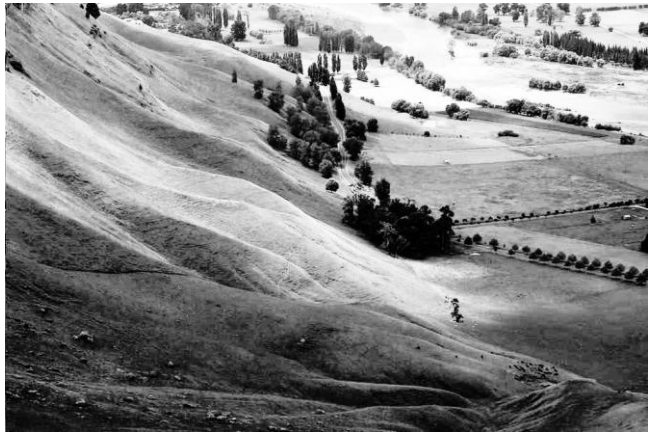
*Assuming uint 8 bit image

Image Preprocessing – Histogram Equalization





1. Find normalized histogram
2. Use the cumulative histogram as the transfer function



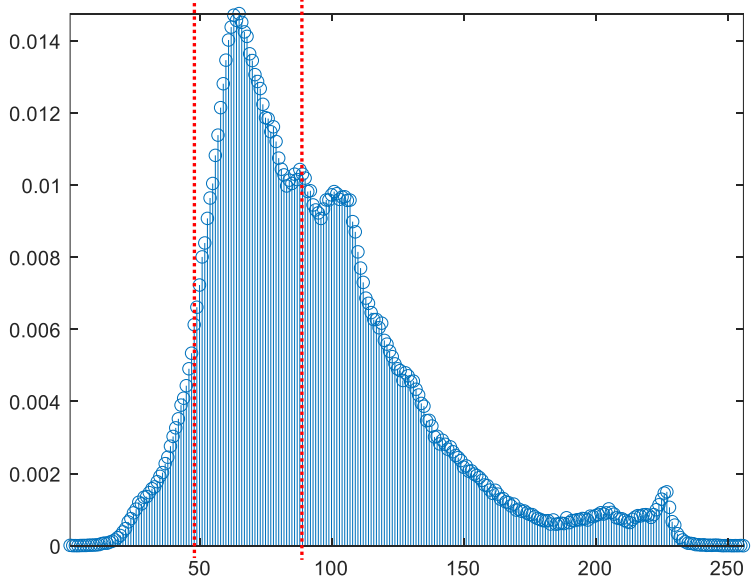
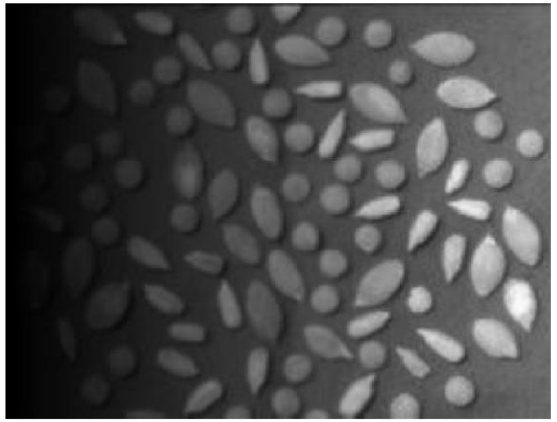
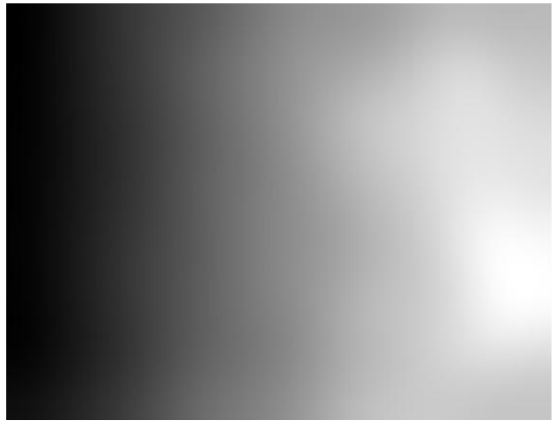


Image Preprocessing - Filtering

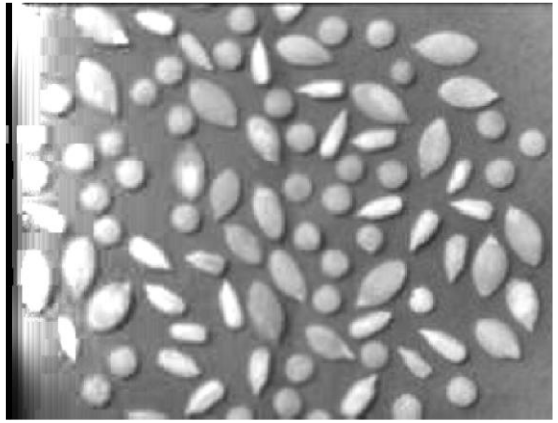
Original



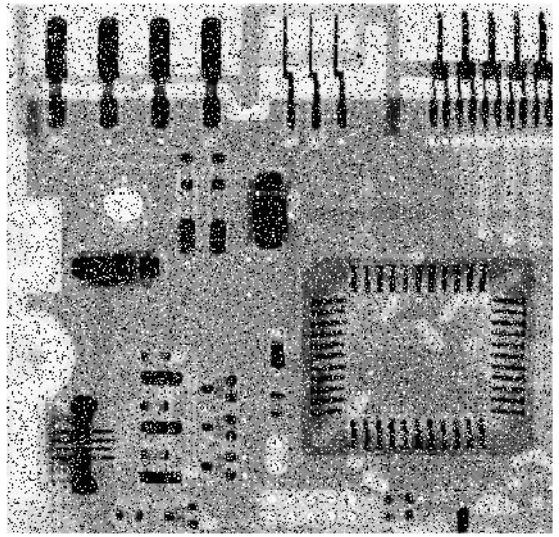
Gaussian Filtered



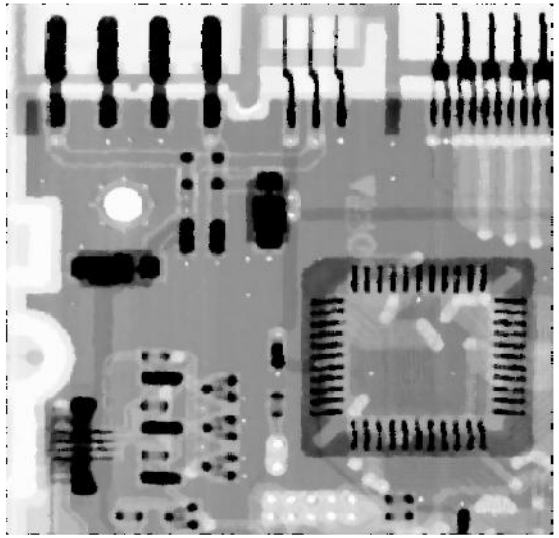
Original ÷ Gaussian Filtered



Original



Median Filtered



Pre-processing is important and could be done on the edge device!

Image Features - Edge Detection (Canny)

Smoothing

1	2	1
2	4	2
1	2	1

Gradient

g_x

-1	1
0	0

g_y

-1	0
1	0

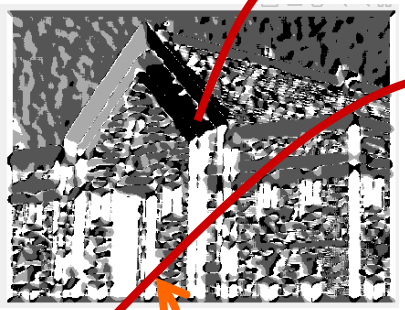
Magnitude

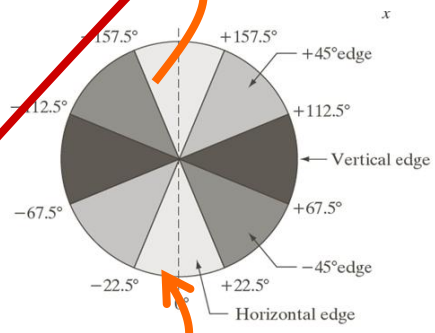
$$M(x,y) = \sqrt{{g_x}^2 + {g_y}^2}$$

Phase

$$\alpha(x,y) = \tan^{-1}\left[\frac{g_y}{g_x}\right]$$

Quantized Phase



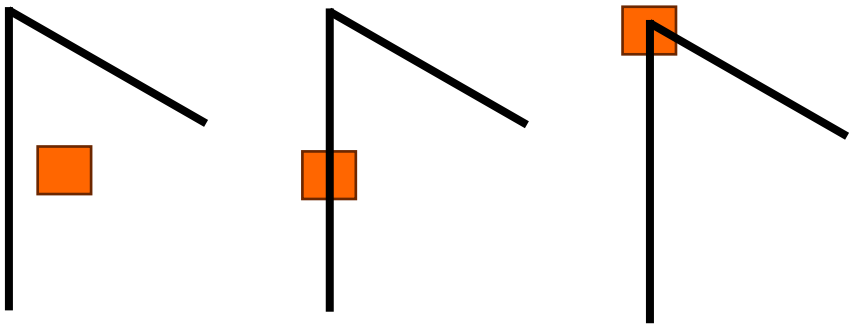


Phase driven Non-maxima Suppression

Double Thresholding

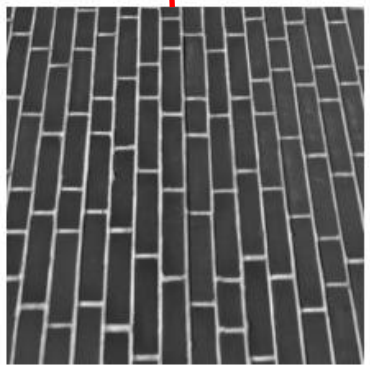
Hysteresis Thresholding

Image Features – (Harris) Corner Detection



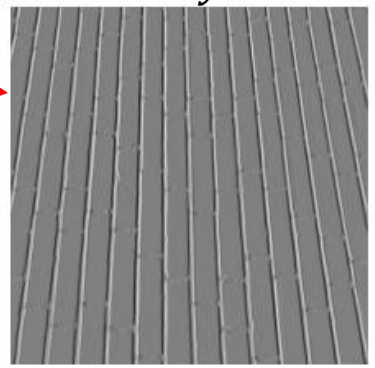
"flat" region "edge" "corner"

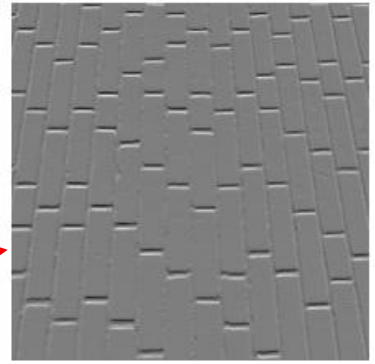
1	0	-1
2	0	-2
1	0	-1



1	2	-1
0	0	0
-1	-2	-1

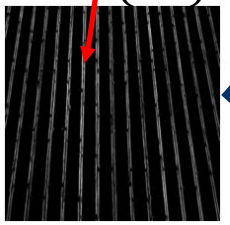
dy



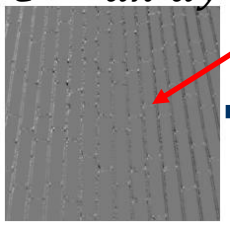


dx

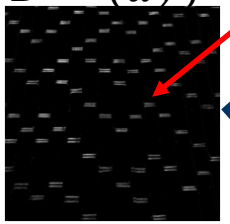
$A = (dx)^2$

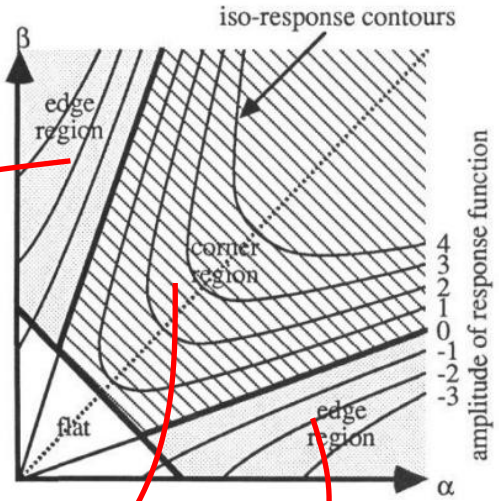


$C = dx\ dy$



$B = (dy)^2$



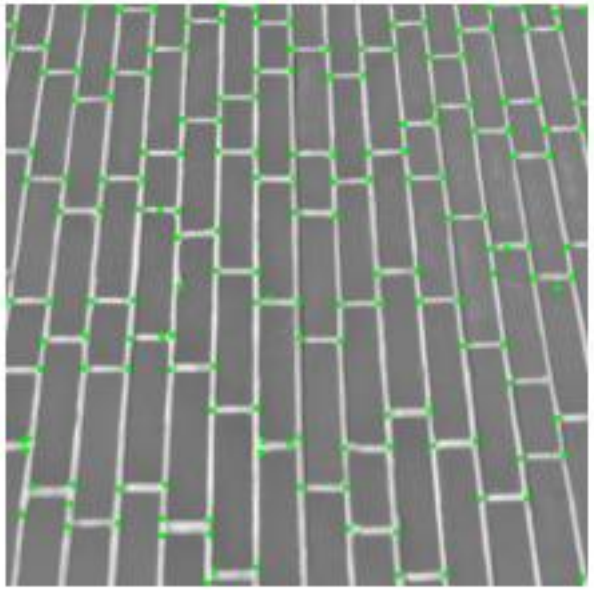


$$\begin{bmatrix} A & C \\ C & B \end{bmatrix}$$

Corner Response Function

$$R = (AB - C^2) - k\,(A + B)^2$$

$$R > Threshold$$



Harris, Chris, and Mike Stephens. "A combined corner and edge detector." In *Alvey vision conference*, vol. 15, no. 50, pp. 10-5244. 1988.

Image Features - Histogram of Oriented Gradients

Smoothing

g_x

-1	1
0	0

g_y

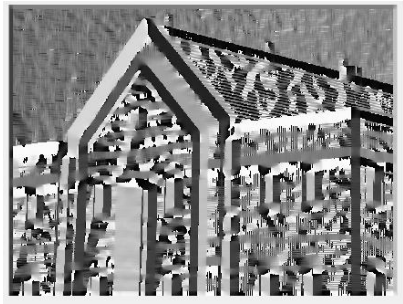
-1	0
1	0

Gradient

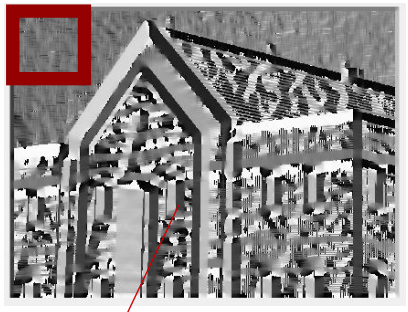
Magnitude

$$M(x,y) = \sqrt{{g_x}^2 + {g_y}^2}$$

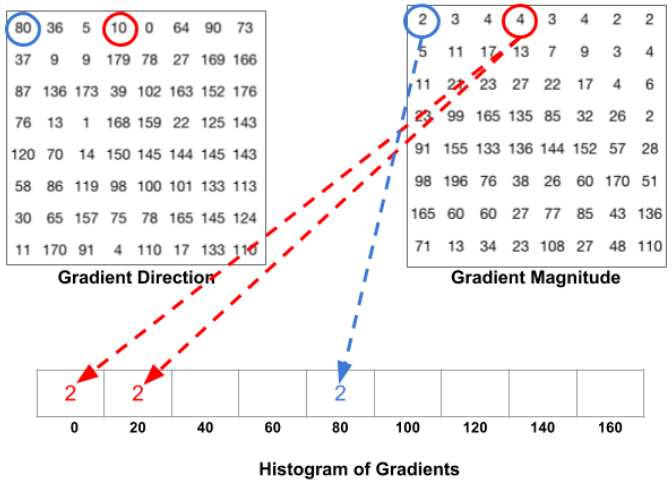
Phase



$$\alpha(x,y) = \tan^{-1}\left[\frac{g_y}{g_x}\right]$$



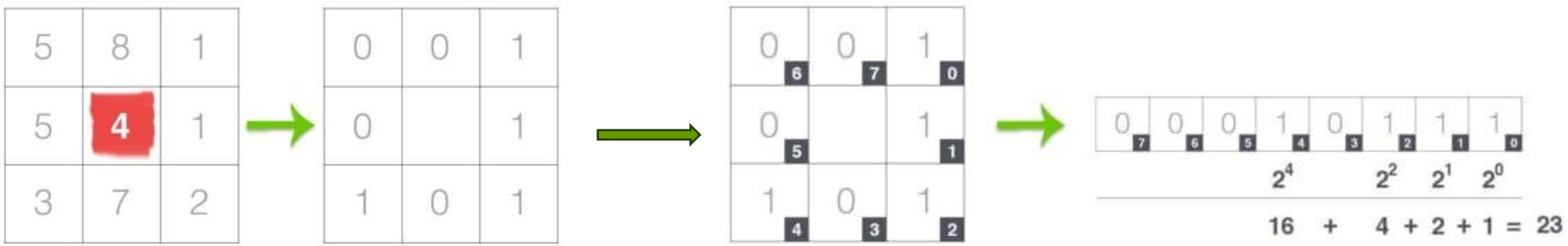
Consider a small window (8x8) and form a histogram for 9 orientations 0, 20, 40, 60, 80, 100, 120, 140, 160 as shown below

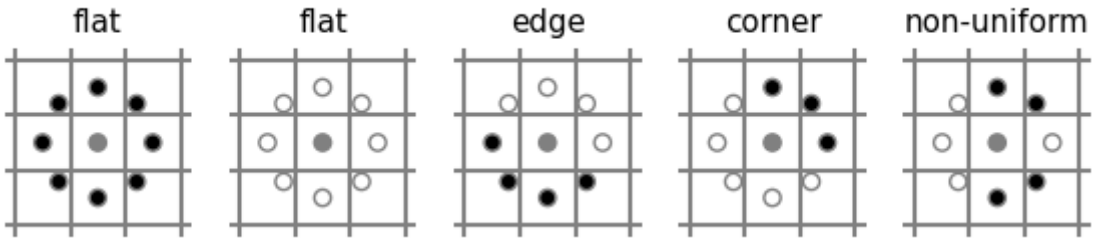


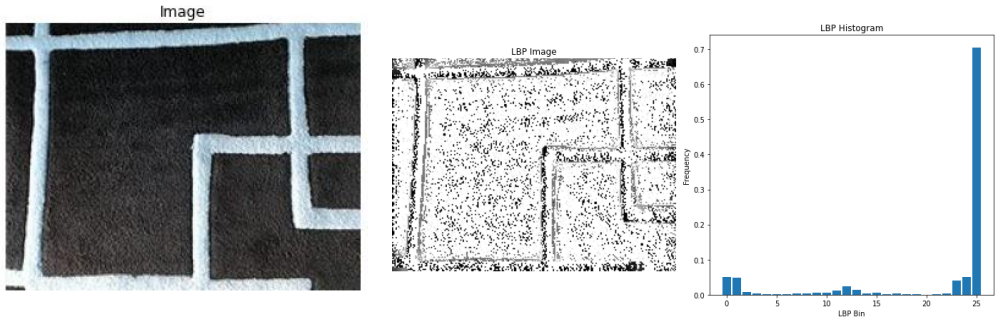
- Normalize the histogram (or before calculating the histogram)
- Concatenate the histograms by moving the window to create a feature vector for the image.

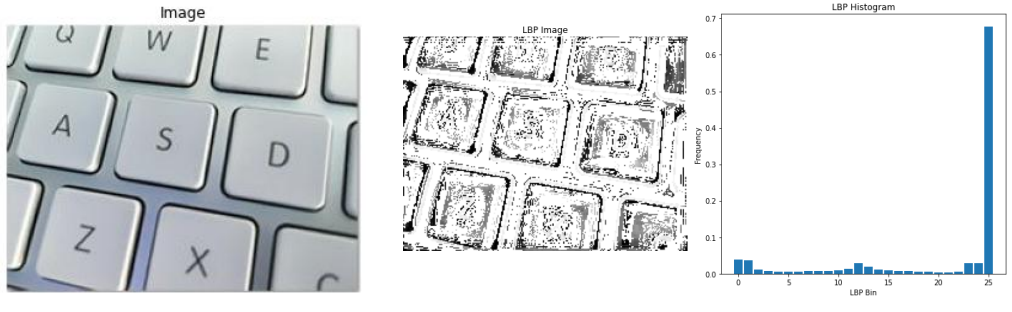
Dalal, N. and Triggs, B., 2005, June. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)* (Vol. 1, pp. 886-893).
https://learnopencv.com/histogram-of-oriented-gradients/

Image Features - Texture Features – Local Binary Pattern (LBP)



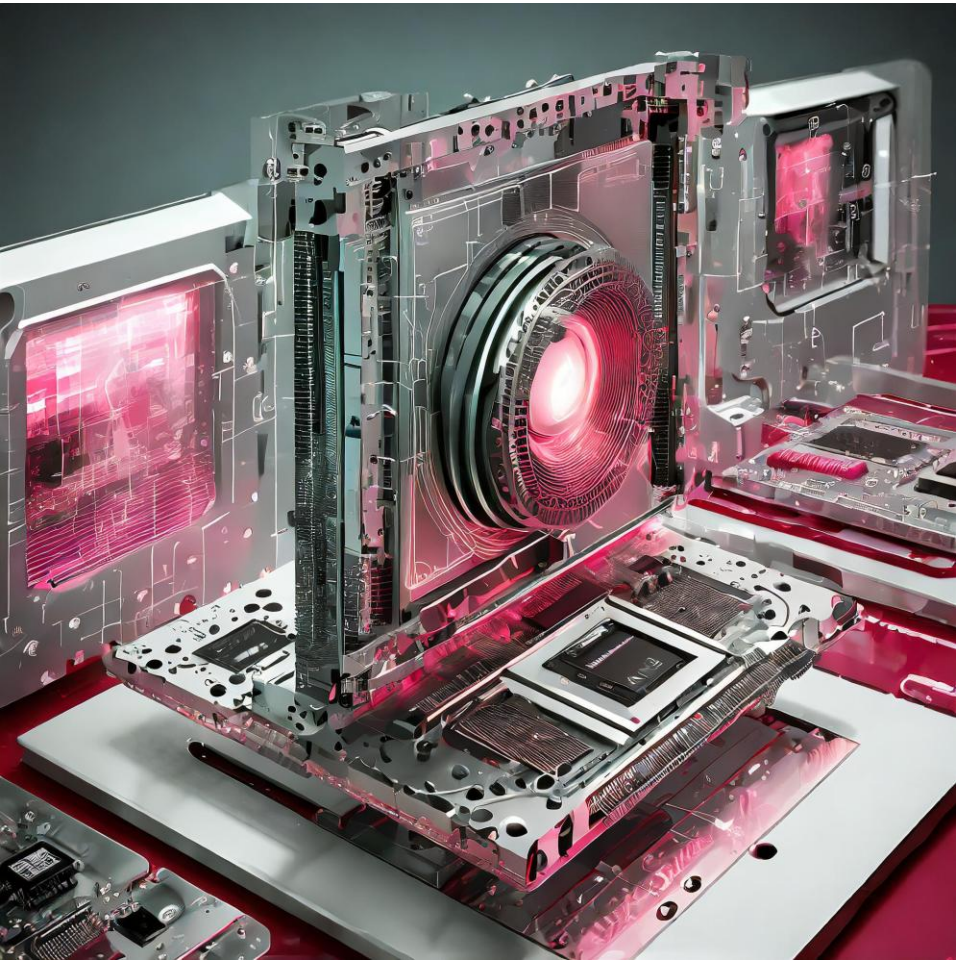






Ojala, T., Pietikainen, M. and Maenpaa, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, *24*(7), pp.971-987.
https://pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/
https://scikit-image.org/docs/stable/auto_examples/features_detection/plot_local_binary_pattern.html

Image to Video Analytics (on Edge)



Limited resources

Real-time requirements with power constraints

Motion estimation and tracking

e.g., video surveillance

Facial/hand landmark detection and gesture control

Video summarization (motion segmentation)

Motion Estimation and Tracking – Optical Flow

Optical flow refers to *the pattern of apparent motion of objects between two consecutive frames* caused by the camera's movement or the objects themselves.

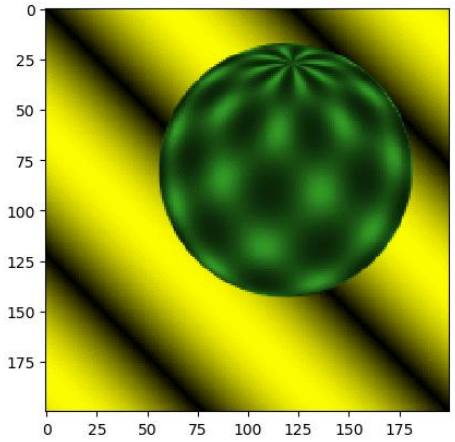
Solution using Lucas-Kanade Method

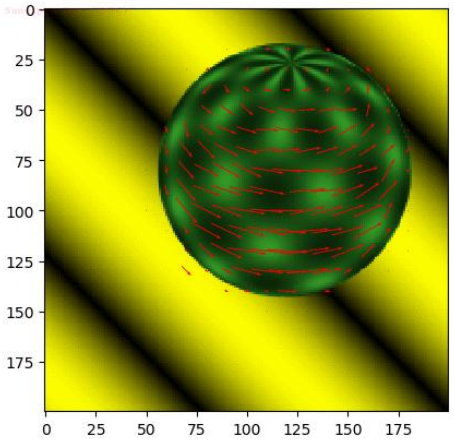
$$I(x, y, t) = I(x + \Delta x, y + \Delta y\,, t + \Delta t)$$

$$Av = b$$

$$A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{bmatrix} \quad v = \begin{bmatrix} v_x \\ v_y \end{bmatrix} \quad b = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{bmatrix}$$

$$\boldsymbol{v} = (\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{A})^{-\boldsymbol{1}}\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{b}$$



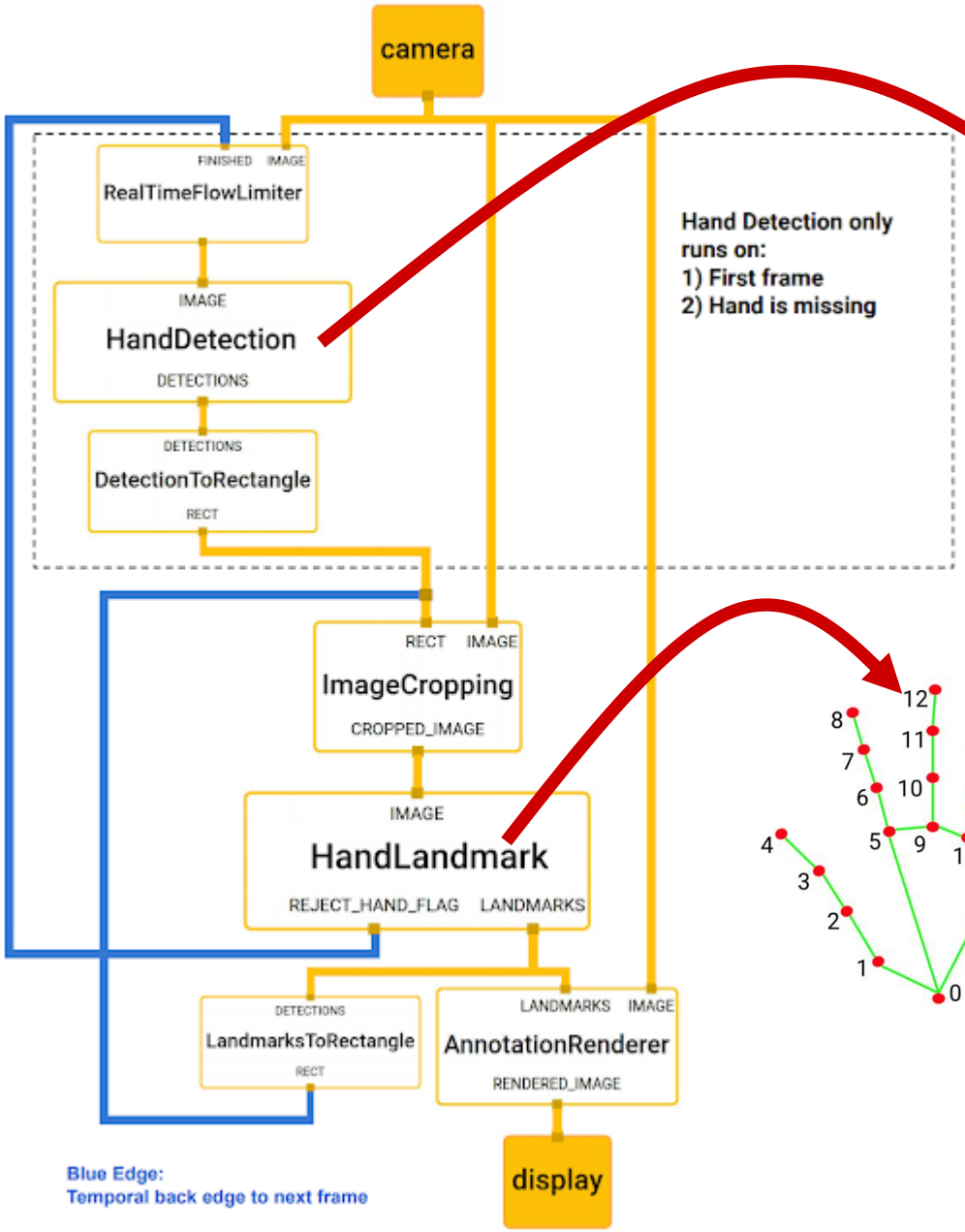


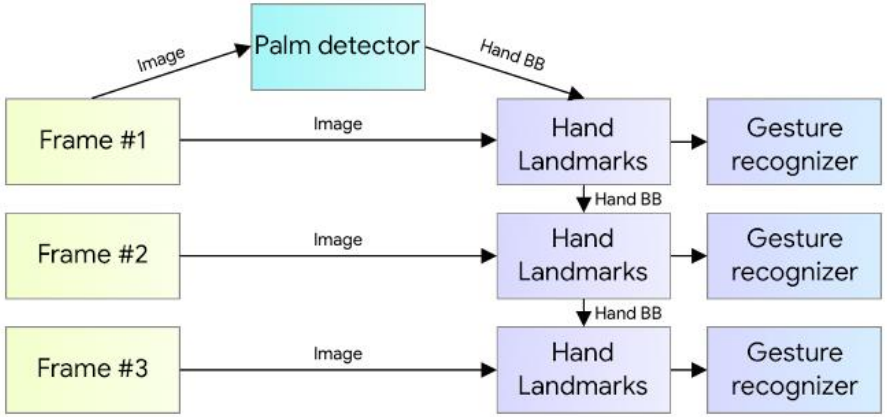
Applications:

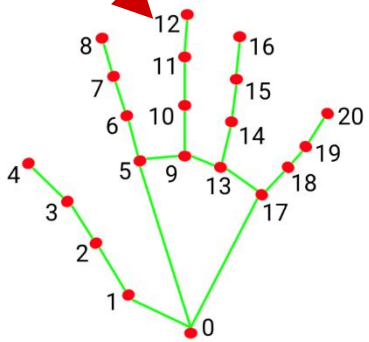
Optical Mouse
Traffic Monitoring
Image Stabilization
Hand/Face tracking

Lucas, Bruce D., and Takeo Kanade. "An iterative image registration technique with an application to stereo vision." In ...nce *intelligence*, vol. 2, pp. 674-679. 1981.
https://www.baeldung.com/cs/optical-flow-lucas-kanade-method
https://sandipanweb.wordpress.com/2018/02/25/implementing-lucas-kanade-optical-flow-algorithm-in-python/

Hand-landmark Detection





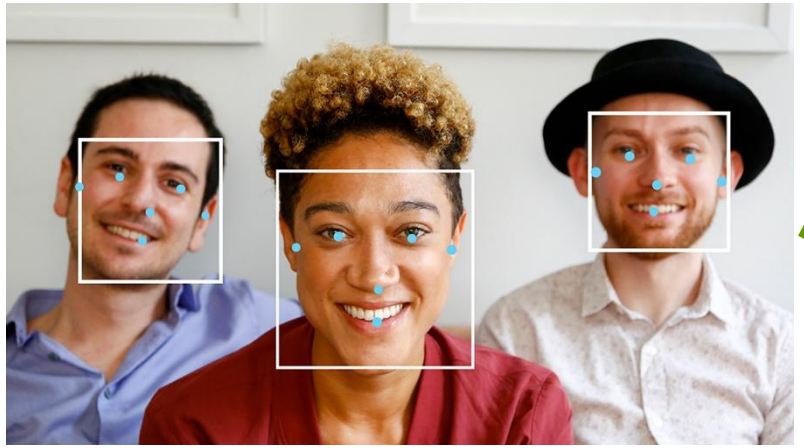


0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

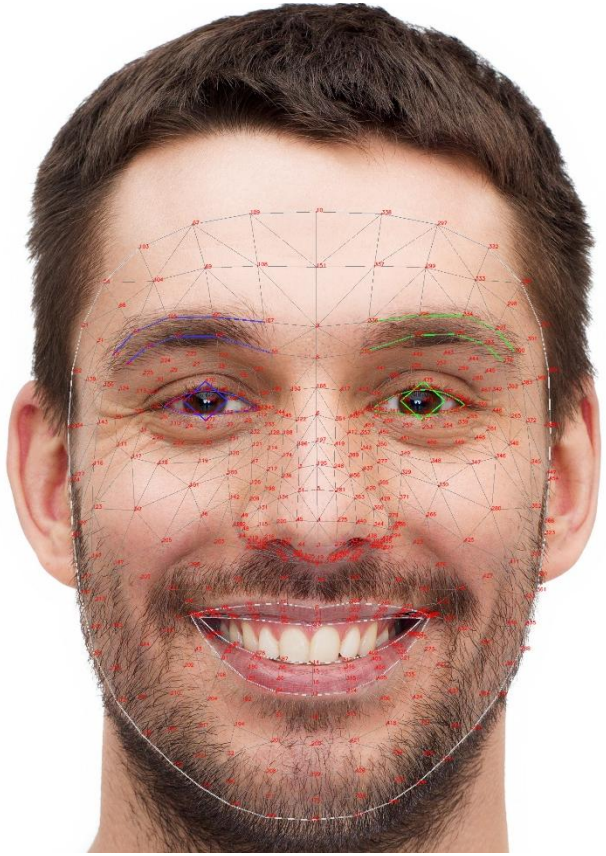
https://blog.research.google/2019/08/on-device-real-time-hand-tracking-with.html

Face-landmark Detection

Face Detection

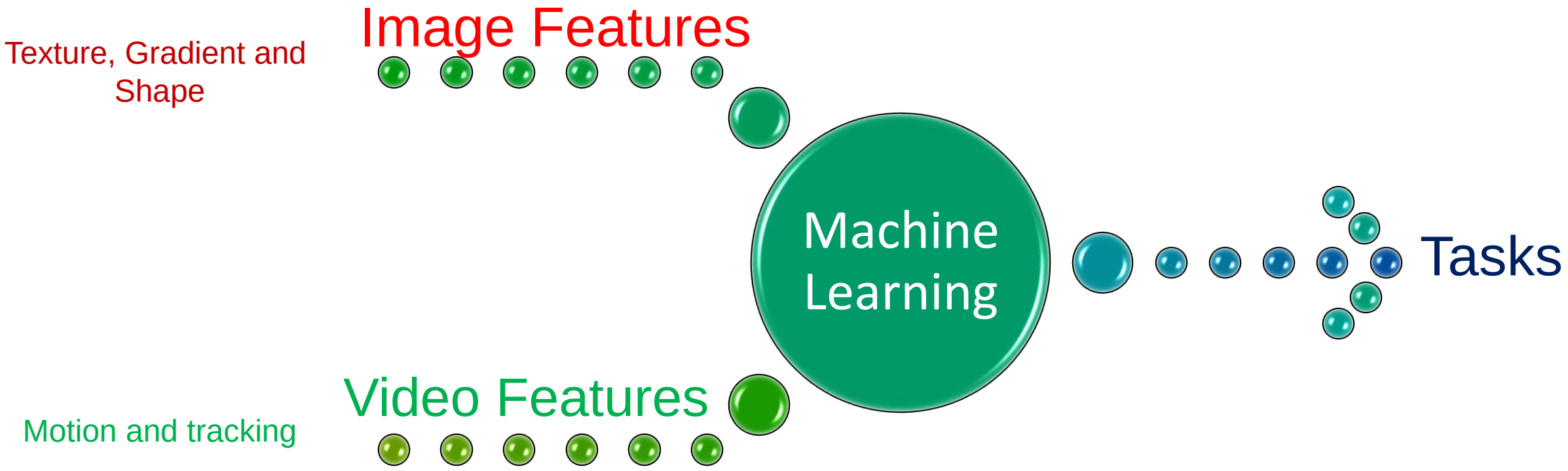


Facial Landmarks



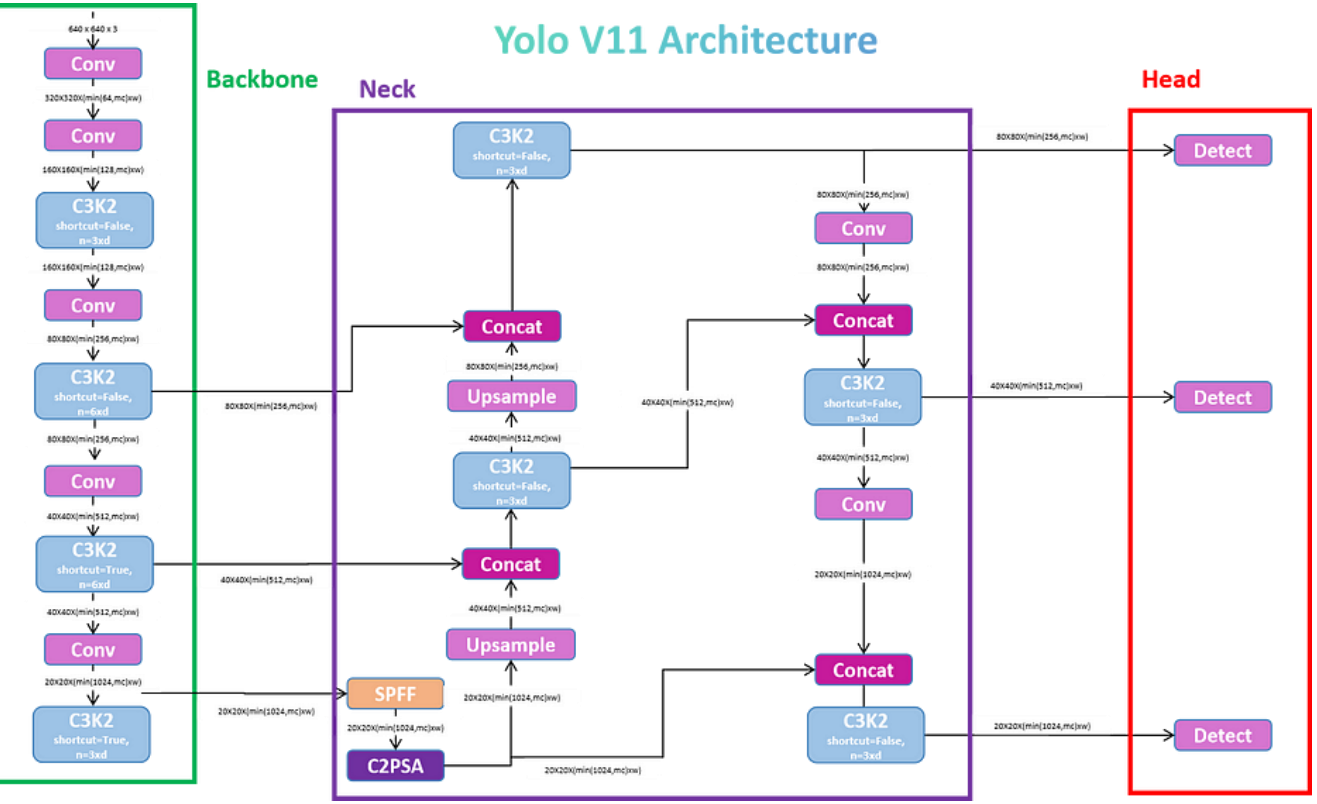
https://developers.google.com/mediapipe/solutions/vision/face_landmarker

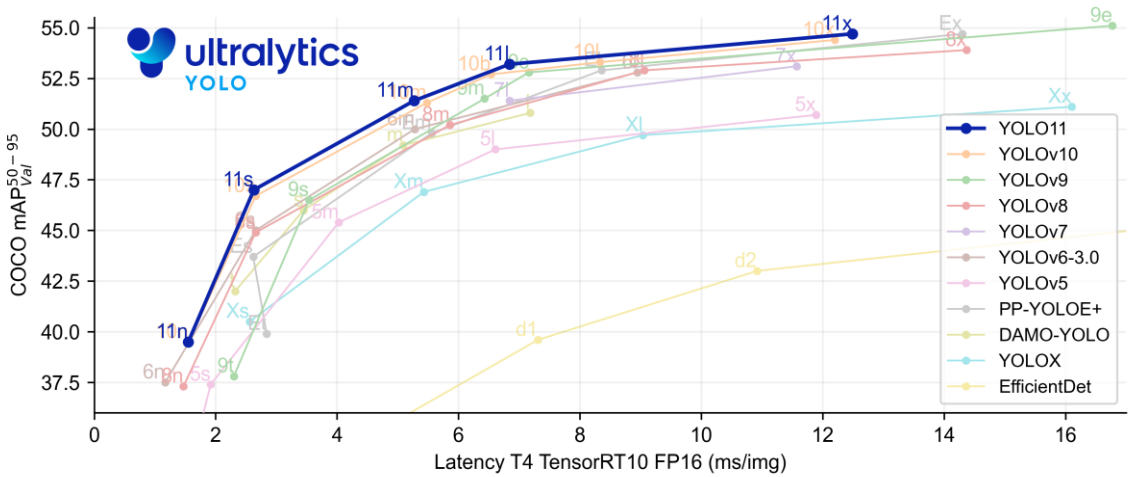
Low Compute Image (Video) Analytics



State-of-the art – YOLOv11

You Only Look Once – version 11 is a lightweight model which can do classification, detection (including oriented detection), instance segmentation and pose estimation!



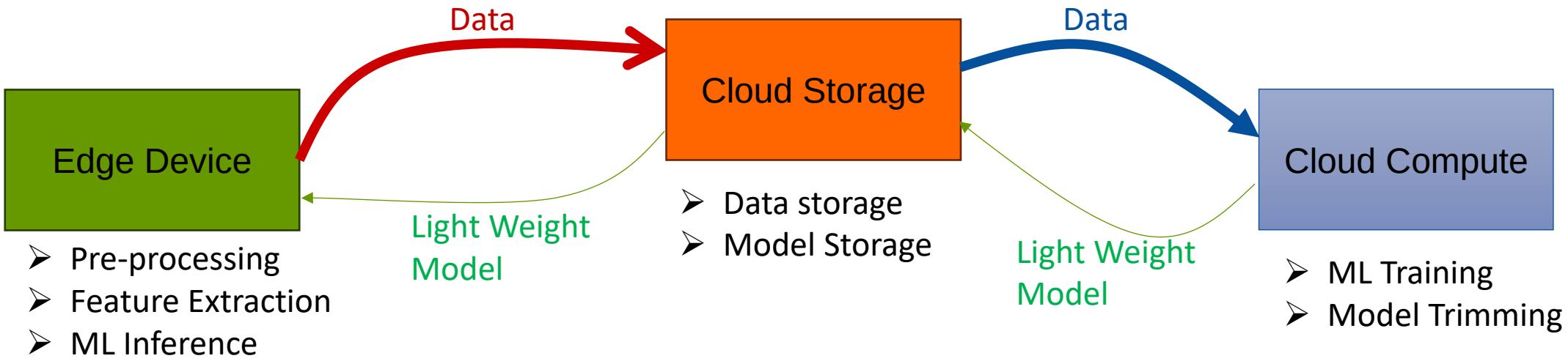


https://medium.com/@nikhil-rao-20/yolov11-explained-next-level-object-detection-with-enhanced-speed-and-accuracy-2dbe2d376f71
https://docs.ultralytics.com/models/yolo11/

Build Your Own Model

- *Pre-processing*
  - *Normalization, Segmentation, Filtering*
- *Feature extraction*
- *Model selection*
  - *CNNs, Transformers*
- *Transfer learning*
  - *Leverage existing models*
- *Useful datasets*
  - *COCO (Common Objects in Context), ImageNet, Open Images Dataset, PASCAL VOC (Visual Object Classes)*
  - *YouTube-8M, UCF-101, Kinetics, HMDB51, Charades*

Current Trend – Cloud to Edge for Scalability

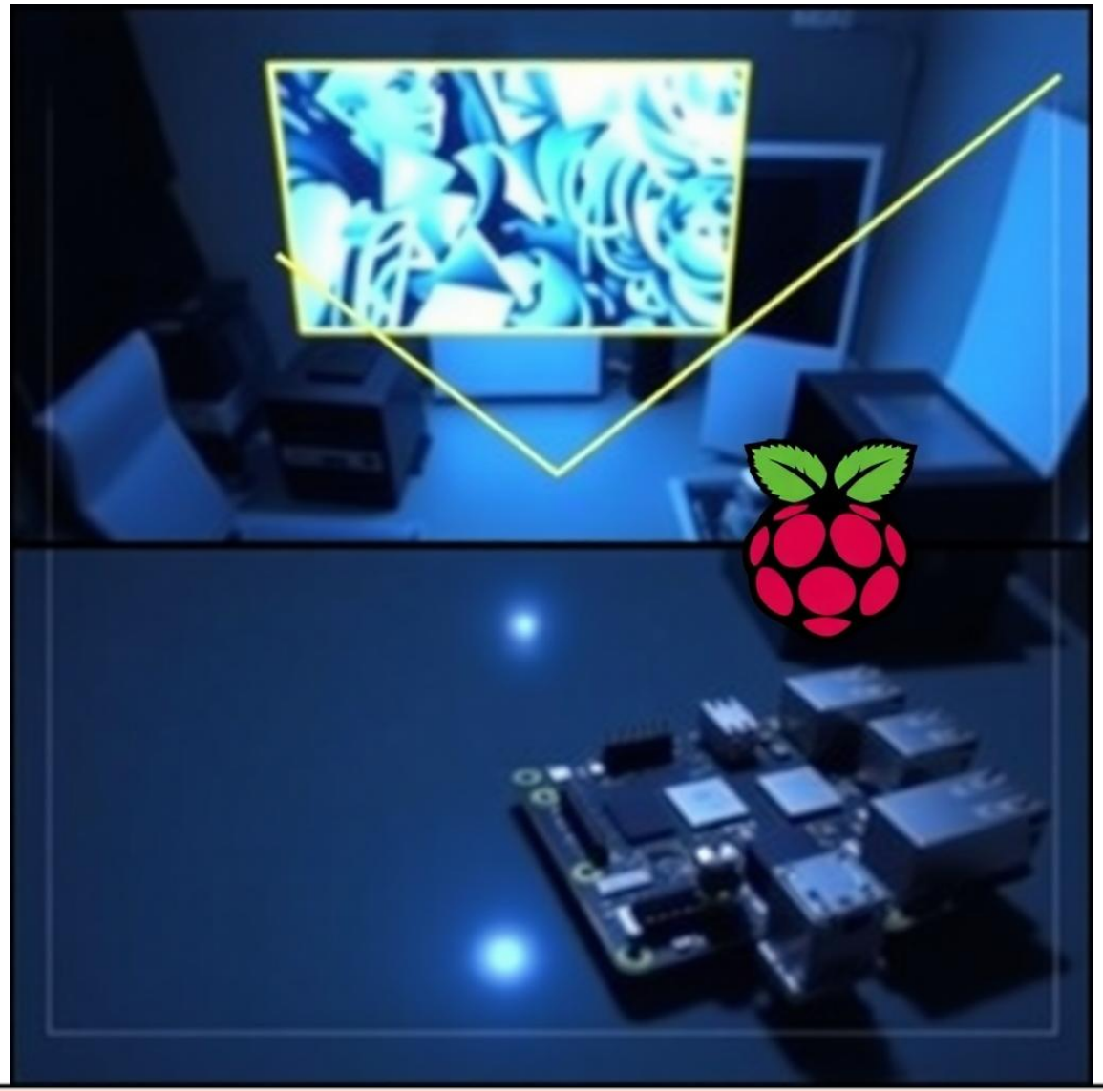


- *Edge Impulse*

- *AWS Rekognition,*
- *Google Cloud Vision API,*
- *IBM Watson Visual Recognition,*
- *Azure AI Vision*

Summary

- *Image/video analytics on edge helps in faster response, data security and enhanced analysis*
- *Pre-processing of images/videos is important*
- *Selection of right input features and model is important*
- *Build in the cloud and adapt to edge for scalability*



Demo of image/video analytics @
https://edgeimageanalytics.streamlit.app/

Contact: mahesh.panicker@singaporetech.edu.sg