******

***AI385***

***Lab 9***

**Image Sequence Analysis**

**Learning Objectives:**

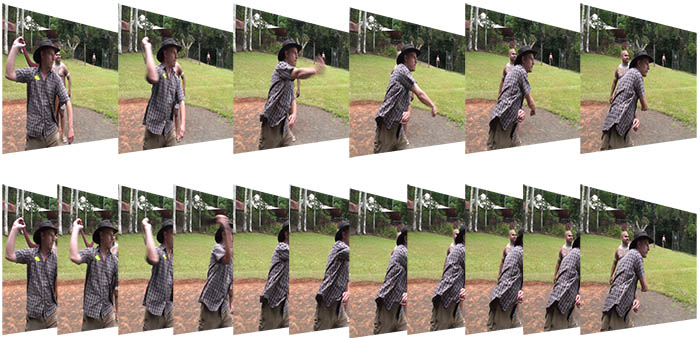
*By the end of this lab, students will be able to:*

* *Understand and implement image sequence analysis using OpenCV in Python.*
* *Apply background subtraction techniques to detect moving objects.*
* *Utilize both manual and built-in OpenCV methods for foreground detection.*
* *Visualize detected objects using bounding boxes and morphological operations.*

***Introduction:***

***What is Image Sequence Analysis?***

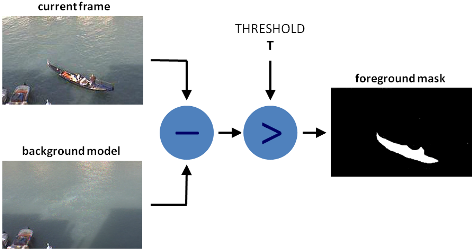
*Image sequence analysis involves processing a series of frames (images from a video or a set of time-based images) to extract meaningful information—such as motion, object detection, or activity tracking.*



***What is Background Subtraction?***

*Background subtraction is a fundamental technique in computer vision used to detect moving objects in videos from static cameras.*

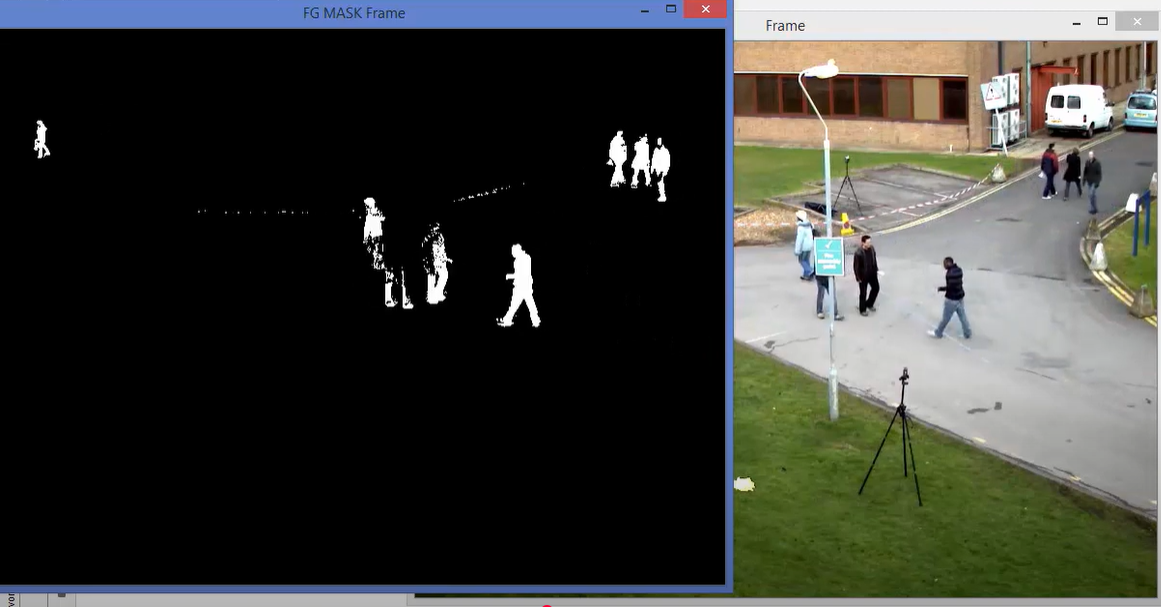
*The basic idea is to compare each new frame to a background model (static or dynamic) and isolate the regions that change over time—these are typically moving objects.*



*There are two major approaches:*

1. ***Manual Background Subtraction:***
   * *Uses a static frame as the background.*
   * *Subtracts new frames from this background.*
   * *Simple and fast but not adaptive to scene changes (e.g., lighting).*
2. ***Gaussian Mixture-based Model (MOG2):***
   * *A more robust and adaptive method.*
   * *Uses multiple Gaussian distributions to model each pixel over time.*
   * *Can handle gradual background changes and detect shadows.*

*This lab explores both techniques to provide a solid understanding of how background subtraction works and its practical applications in surveillance, motion detection, and object tracking.*

**

**Exercise 1: Manual Background Subtraction for Object Detection**

**Objective:**

Detect moving objects by subtracting each frame from a static background frame.

**Code:**

#Read video or image sequence using:

cap = cv.VideoCapture('video.mp4')

#Capture the first frame as the static background:

ret, background = cap.read()

#Subtract the current frame from the background:

diff = cv.subtract(background, current\_frame)

#Apply thresholding to highlight changes:

ret, thresh = cv.threshold(diff, 20, 255, cv.THRESH\_BINARY)

#Clean the binary mask using morphological closing:

kernel = np.ones((5, 5), np.uint8)

closing = cv.morphologyEx(thresh, cv.MORPH\_CLOSE, kernel)

#Draw bounding boxes around detected objects:

coords = cv.findNonZero(closing)

x, y, w, h = cv.boundingRect(coords)

cv.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

**Exercise 2: Object Detection Using MOG2 (OpenCV Built-in)**

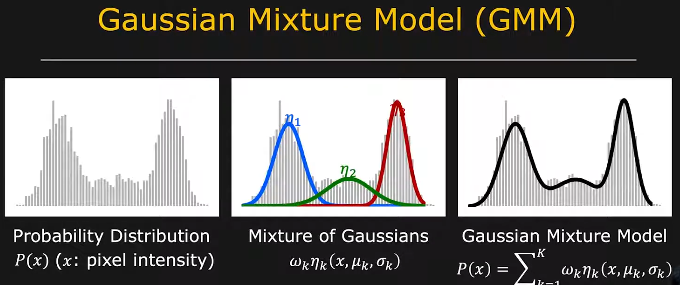
**Objective:**

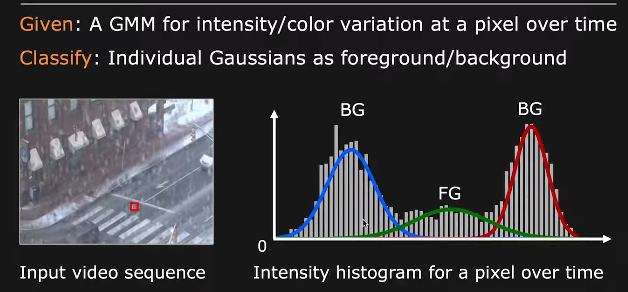
Detect moving objects using OpenCV’s built-in MOG2 background subtractor for dynamic environments.

**How MOG2 Works (Simplified):**

* Each pixel is modeled as a mixture of Gaussians over time.
* Frequently observed colors form the background.
* Unusual colors (outliers) are flagged as foreground (moving objects).
* A graph of two people

  AI-generated content may be incorrect.This method adapts to scene changes, making it suitable for real-world environments.





A collage of a person walking on a road

AI-generated content may be incorrect.

**Code:**

#Create the background subtractor object:

backSub = cv.createBackgroundSubtractorMOG2(history=100, varThreshold=40, detectShadows=True)

#Apply the subtractor to each frame:

fg\_mask = backSub.apply(frame)

#Display the result:

cv.imshow('Foreground Mask', fg\_mask)

#Post-process and draw contours or bounding boxes as in Exercise 1.

**For more details:**

* + 1. <https://www.simonwenkel.com/notes/software_libraries/opencv/background-subtraction-using-opencv.html>
    2. Gaussian Mixture-based Background/Foreground Segmentation Algorithm

<https://www.youtube.com/watch?v=0nz8JMyFF14&ab_channel=FirstPrinciplesofComputerVision>