LOYOLA INSTITUTE OF TECHNOLOGY AND SCIENCE

DEPARTMENT OF INFORMATION TECHNOLOGY

COMPLETED THE PROJECT NAMED AS

Motion Tracking On Video OpenCV

Submitted by,

Team Members

1. S.ROJAPOO (961222205044)
2. K.R.SREENTHI(961222205050)
3. E.SUMITHRA(961222205053)

AIM:

DETECT AND TRACK MOVING OBJECTS IN A VIDEO USING OPENCV.

Abstract:

Motion tracking is a critical component in various computer vision applications, including surveillance, human-computer interaction, and activity recognition. This project explores the implementation of motion tracking on video using OpenCV, a widely-used open-source computer vision library. The primary objective is to detect and track moving objects in video sequences through background subtraction and contour detection. By applying techniques such as frame differencing and using the BackgroundSubtractorMOG2 algorithm, moving objects are identified in each frame. These objects are then enclosed in bounding boxes to visualize their motion paths. The system operates in real-time and is capable of filtering out noise through contour area thresholds. This approach provides a foundational method for motion analysis and can be extended with

advanced tracking algorithms like CSRT or integrated with object detection frameworks such as YOLO for enhanced performance.

Introduction:

Motion tracking is a key technique in computer vision that allows us to detect and follow moving objects in a video stream. This project utilizes OpenCV, an open-source computer vision library, to implement motion tracking by identifying changes between frames.

System Architecture:

Motion Tracking on Video using OpenCV

1. Input Module

Video Source: Live camera feed or pre-recorded video file.

Frame Extraction: Captures individual frames from the video.

2. Preprocessing Module

Resize & Grayscale Conversion: Standardize frame size and convert to grayscale for processing.

Noise Reduction: Apply Gaussian blur or morphological operations to reduce noise.

3. Motion Detection Module

Background Subtraction: Use algorithms like cv2.createBackgroundSubtractorMOG2() to isolate moving areas.

Frame Differencing (optional): Detect changes between consecutive frames.

4. Contour Detection Module

Thresholding & Dilation: Highlight motion areas.

Find Contours: Extract object outlines using cv2.findContours().

Filter Contours: Remove small/noise contours using area thresholds.

5. Tracking Module

Bounding Box Drawing: Enclose detected motion regions with rectangles.

Tracker (Optional): Use OpenCV trackers (e.g., CSRT, KCF) for object-specific tracking.

6. Visualization Module

Overlay Results: Draw bounding boxes and labels on the original frame.

Display Output: Show the processed video in real-time using cv2.imshow().

7. Output Module (Optional)

Recording/Logging: Save tracked video or log object movements.

Alerts/Integration: Interface with other systems for alerts or further processing.

Code structure

1.Frame Differencing

We use the difference between consecutive frames to detect motion:

```
diff = cv2.absdiff(frame1, frame2)
```

output

```
+----------------------+
| |
| White blob | <-- Motion difference
| |
+----------------------+
```

2 .Thresholding and Dilation

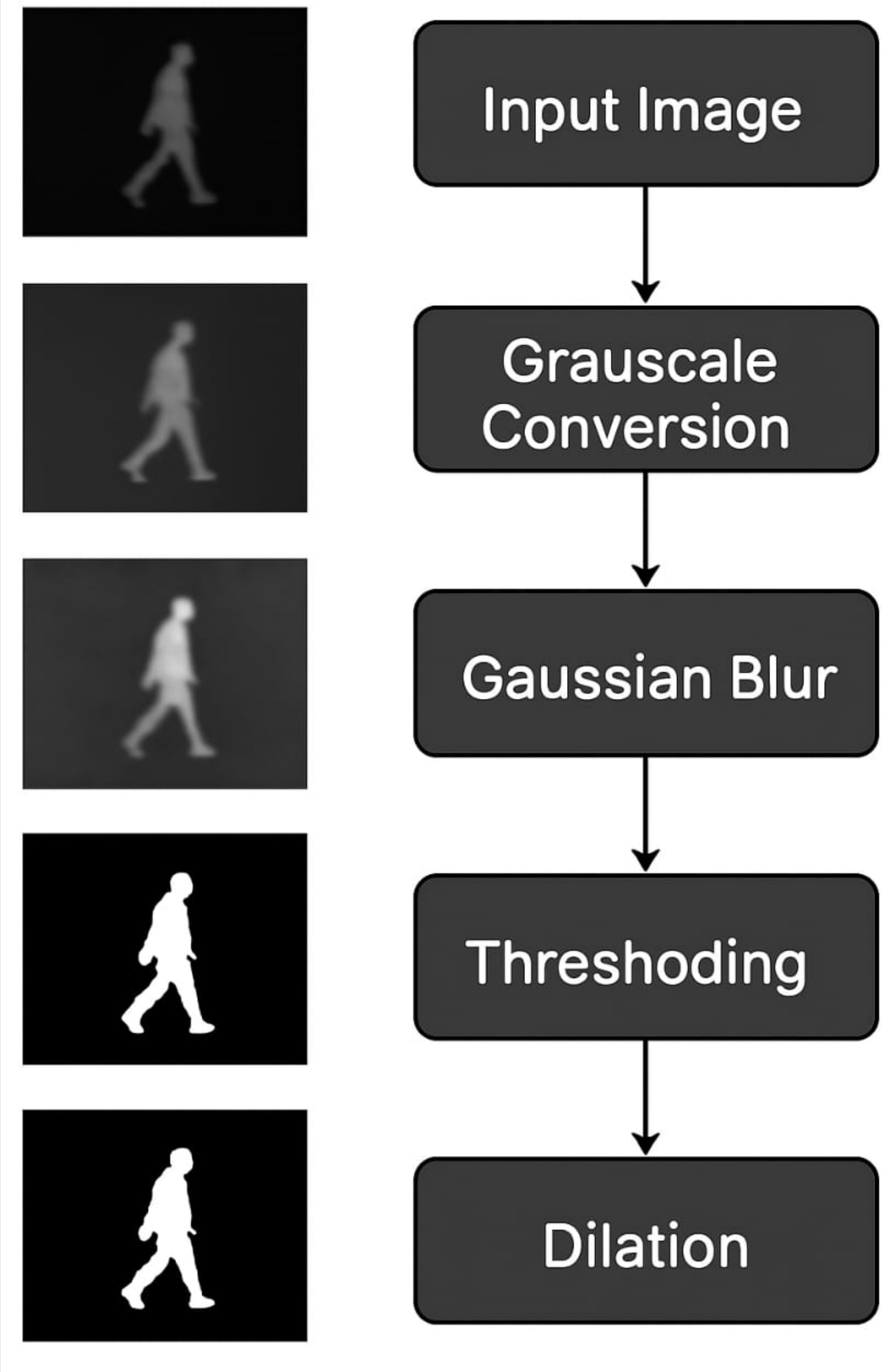
```
gray = cv2.cvtColor(diff, cv2.COLOR_BGR2GRAY)

blur = cv2.GaussianBlur(gray, (5,5), 0)

_, thresh = cv2.threshold(blur, 20, 255,
cv2.THRESH_BINARY)

dilated = cv2.dilate(thresh, None, iterations=3)
```

Output:

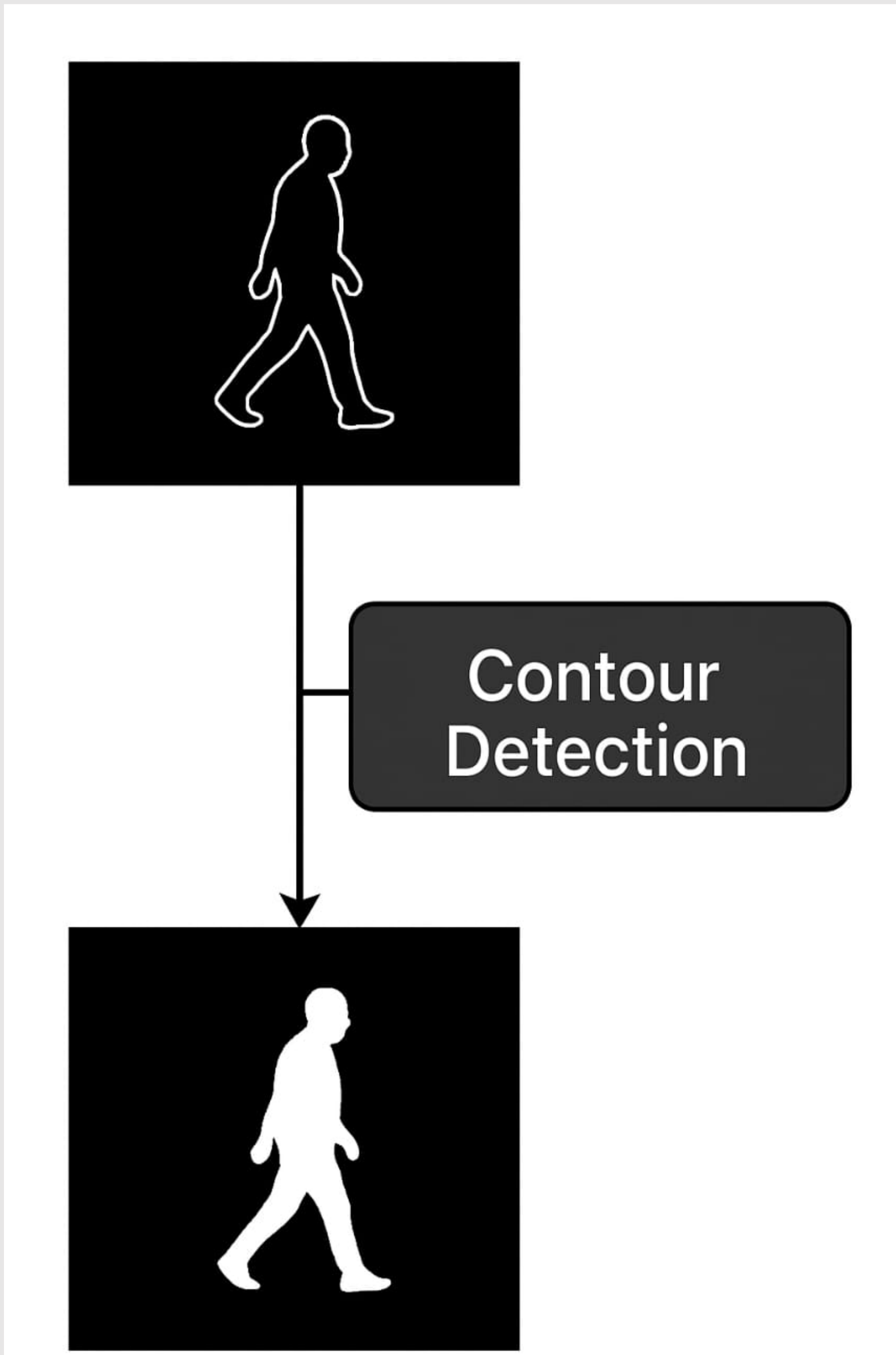


3 .Contour Detection

Find contours to detect motion regions:

```
contours, _ = cv2.findContours(dilated, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
```

Output:

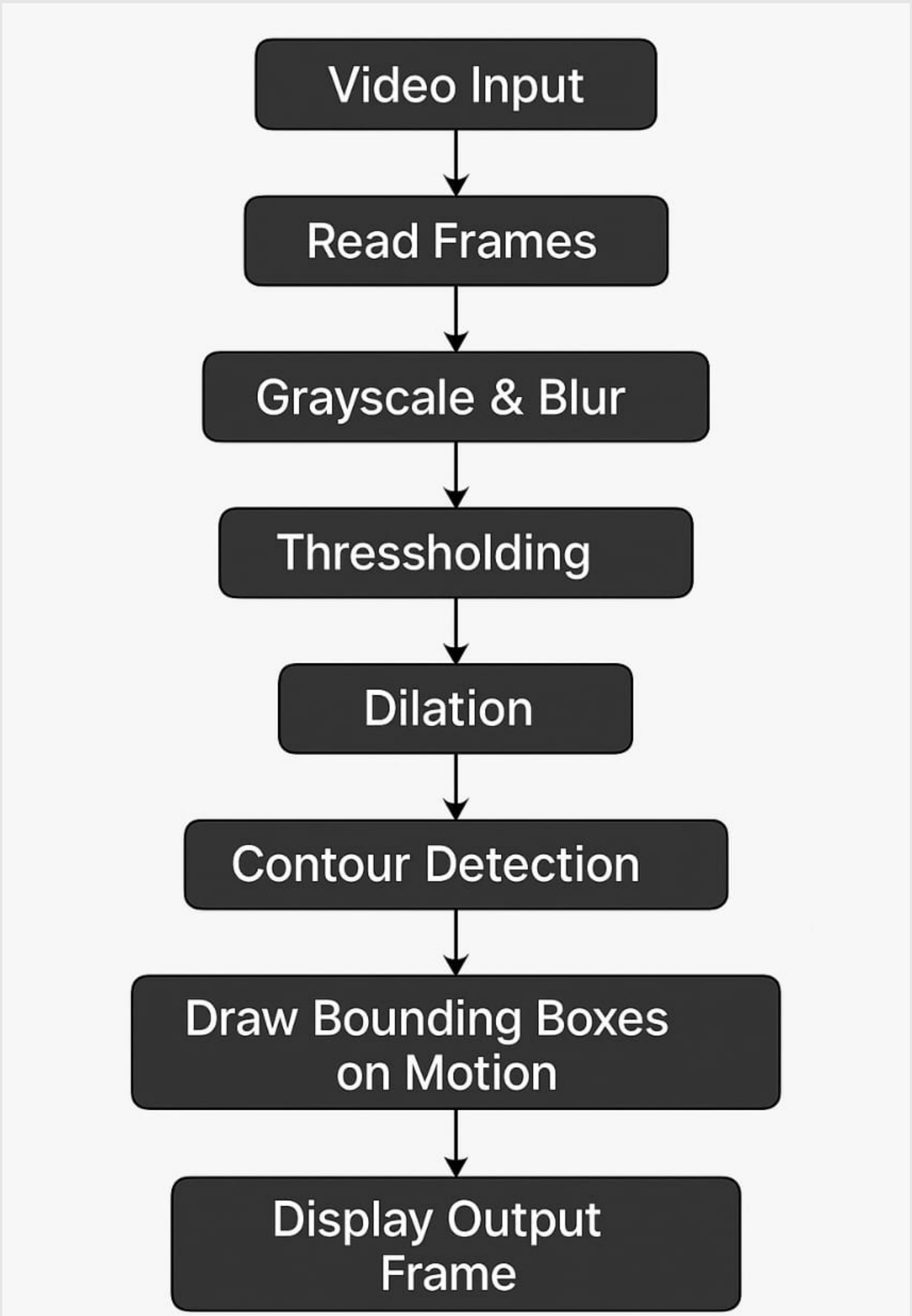


4. Drawing Bounding Boxes

Draw rectangles around detected moving objects:

```
for contour in contours:
 if cv2.contourArea(contour) < 700:
 continue
 x, y, w, h = cv2.boundingRect(contour)
 cv2.rectangle(frame1, (x, y), (x+w, y+h), (0, 255, 0), 2)
```

Output:


Video Input
Read Frames
Grayscale & Blur
Thressholding
Dilation
Contour Detection
Draw Bounding Boxes on Motion
Display Output Frame

OBJECTIVE

The Main Objective Of This Project Is To:

Detect Motion In Video Frames.

Track Moving Objects Using Bounding Boxes.

Visualize The Motion Path Over Time.

TOOLS AND TECHNOLOGIES

Language: Python

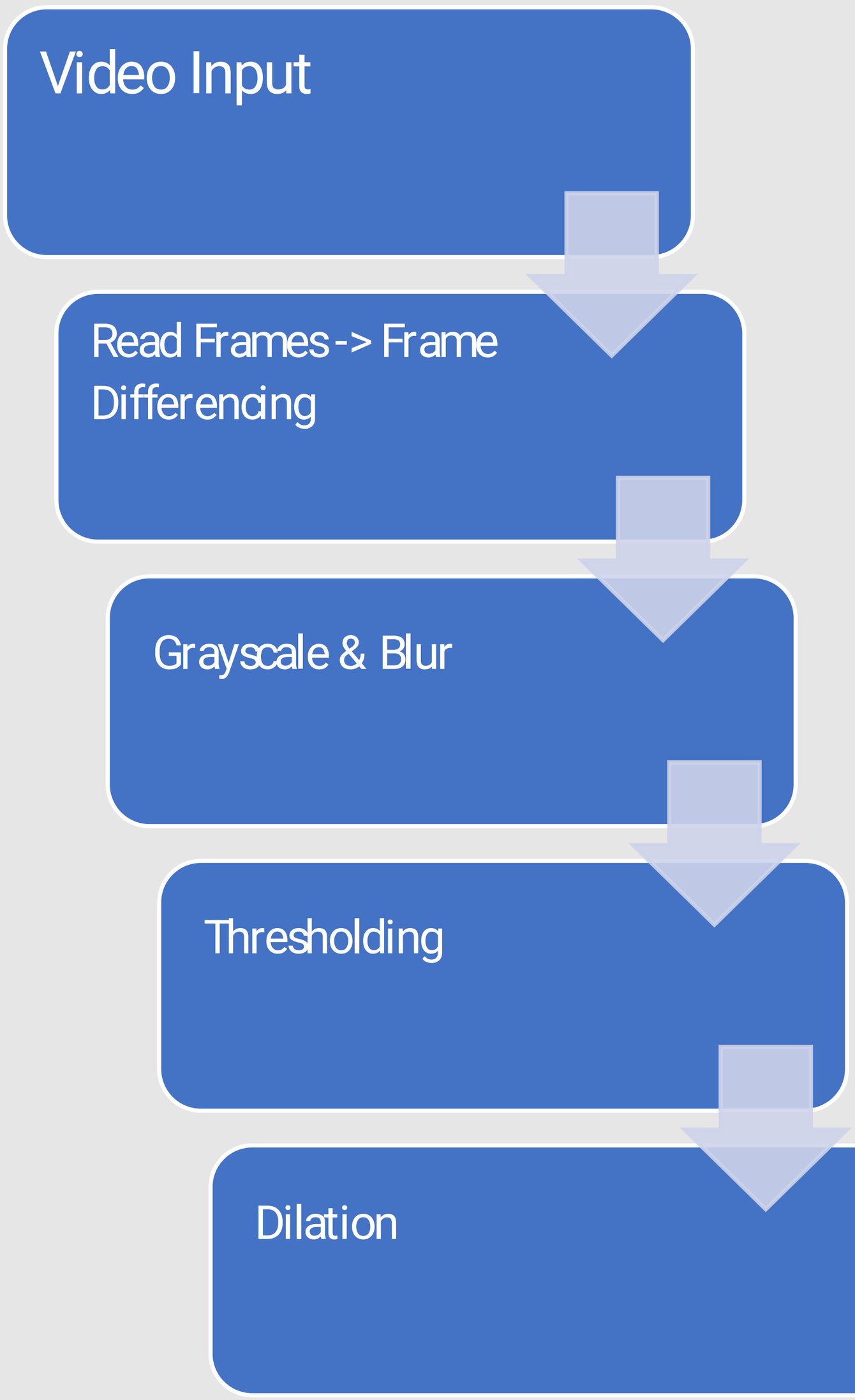
Library: Opencv (Cv2)

Ide: Jupyter Notebook / Vs Code

Video Source: Webcam Or Pre-Recorded Video

DIAGRAM

HERE'S A SIMPLIFIED FLOWCHART OF THE PROCESS:



CONCLUSION

This Project Demonstrates How Opencv Can Be Effectively Used For Motion Detection And Tracking In Video Feeds. By Leveraging Frame Differencing, Thresholding, And Contour Analysis, We Successfully Track Movement In Real-Time. This Approach Has Applications In Surveillance, Activity Recognition, And Human-Computer Interaction.