S. B. JAIN INSTITUTE OF TECHNOLOGY, MANAGEMENT & RESEARCH, NAGPUR.

Practical No. 11

Aim: Design and Implementation of a Background Subtraction System Robust to Lighting Changes, Shadows, and Dynamic Scenes.

Name of Student: Shrutika Pradeep Bagdi

Roll No.: CS22130

Semester/Year: VII/IV

Academic Session: 2025-2026

Date of Performance:

Date of Submission:

AIM: Design and Implementation of a Background Subtraction System Robust to Lighting Changes, Shadows, and Dynamic Scenes

OBJECTIVE/EXPECTED LEARNING OUTCOME:

The objectives and expected learning outcome of this practical are:

- To comprehend the fundamental difference between simple background subtraction (e.g., simple averaging) and adaptive/probabilistic models (e.g., MOG, ViBe).
- To successfully implement and configure at least two distinct adaptive background subtraction algorithms (MOG2 and an advanced method).

THEORY:

1.Background Subtraction

Background subtraction is a fundamental technique in computer vision for detecting moving objects in video sequences. It models the background and classifies pixels that deviate significantly as foreground.

- **MOG2 (Mixture of Gaussians 2):** Each pixel is modeled as a mixture of Gaussians to capture variations (e.g., moving leaves). Foreground is detected when pixel values deviate from the dominant Gaussian distribution.
- **Parameters:**
  - *history*: number of frames considered in the background model.
  - *varThreshold*: sensitivity to differences between foreground and background.
  - *detectShadows*: whether to classify soft differences as shadows (value 127 in mask).

2. Illumination Normalization (CLAHE)

Lighting variations affect detection accuracy. **Contrast Limited Adaptive Histogram Equalization (CLAHE)** enhances local contrast by applying histogram equalization on small tiles in the image.

- Applied on the V (brightness) channel in HSV color space.
- Prevents over-amplification of noise.
- Helps maintain stable background/foreground separation under changing light conditions.

3. Shadow Detection and Removal

Shadows are often misclassified as foreground. To address this:

1. **MOG2 built-in shadow detection**: Shadows are labeled as 127 in the mask.
2. **HSV-based heuristic filter**: Shadows usually have lower intensity (V channel) but similar hue and saturation. A threshold on relative brightness is used to remove dark regions incorrectly classified as objects.

4. Postprocessing

Foreground masks contain noise (small blobs, holes). To refine masks:

- **Median Blur:** Removes salt-and-pepper noise.
- **Morphological Opening:** Removes small false positives.
- **Morphological Closing:** Fills small holes inside detected objects.
- **Thresholding:** Ensures binary mask (0 for background, 255 for foreground).

5 Contour Analysis

Contours of connected regions are extracted from the cleaned mask. Small contours (area < threshold) are removed. Bounding boxes are drawn around valid objects for visualization.

- Provides object locations and approximate sizes.
- Useful for object tracking and higher-level tasks.

6 Performance Evaluation

The system performance is evaluated qualitatively (visual results) and quantitatively (frames per second).

- **Visualization:** Overlays bounding boxes and translucent masks on video frames.
- **Efficiency:** FPS is displayed to monitor real-time capability.

Algorithm:

Step 1: Start

Step 2: Input Video Source

- Accept input from either a video file or webcam.
- Initialize video capture object.

Step 3: Initialize Parameters

- Set configuration values:
  - Background subtractor parameters (history, threshold, shadow detection).
  - CLAHE parameters for illumination normalization.
  - Morphological kernel sizes.
  - Minimum contour area threshold.

Step 4: Preprocessing (Optional)

- Convert each frame from BGR → HSV.
- Apply **CLAHE** on V (brightness) channel to handle illumination variation.
- Reconstruct frame in BGR.

Step 5: Background Subtraction

- Apply **MOG2 subtractor** to obtain raw foreground mask.
- If enabled, remove shadow pixels (value = 127).

Step 6: Shadow Filtering (Optional)

- Convert frame to HSV.
- Identify pixels with significantly lower brightness (V) than mean.
- Remove those pixels from mask (assumed shadows).

Step 7: Postprocessing

- Apply **median blur** to reduce noise.
- Apply **morphological opening** to remove small false detections.
- Apply **morphological closing** to fill small holes.
- Threshold to obtain binary mask (0/255).

Step 8: Contour Detection

- Extract contours from cleaned mask.
- Filter contours based on minimum area.
- For each valid contour:
  - Draw bounding box on original frame.
  - Mark detection area and store mask.

Step 9: Visualization & Output

- Overlay binary mask on original frame (green foreground highlight).
- Display detection count and real-time FPS.

- Show windows if enabled.
- Save processed video and mask outputs if enabled.

Step 10: Repeat Steps 4–9 for all frames until end of video or ESC key pressed.

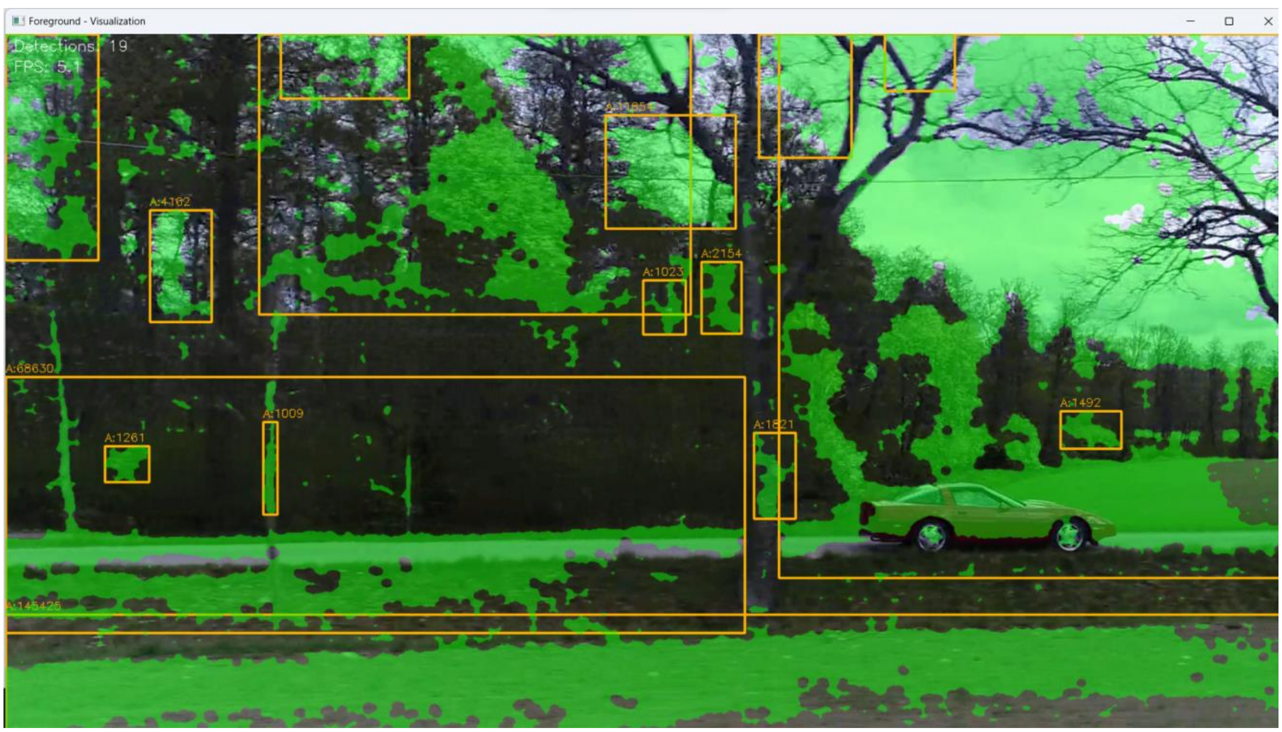
Step 11: Release Resources

- Release video capture and writers.
- Close all OpenCV windows.
- Print processing summary (frames, time, FPS).

Step 12: Stop

Code:

INPUT & OUTPUT:



CONCLUSION:

This practical successfully demonstrates how background subtraction with illumination normalization and shadow removal can accurately detect moving objects in dynamic video scenes.

DISCUSSION QUESTIONS:

1. How does MOG2 background subtractor differ from simple frame differencing?
2. What is the role of CLAHE in handling illumination changes?
3. How are shadows detected and removed in HSV color space?
4. Why are morphological operations important after background subtraction?
5. What are the trade-offs when setting minimum contour area threshold?

REFERENCES:

- Computer Vision: Algorithms and Applications, Richard Szeliski, 2010, Springer.
- Computer Vision - A modern Approach, D. Forsyth, J. Ponce, 2nd Edition 2011, Pearson India.
- OpenCV Computer Vision with Python, Joseph Howse, 2013, Packt Publishing.
- Dictionary of Computer Vision and Image Processing, R. B. Fisher, T. P. Breckon, K. Dawson-Howe, A. Fitzgibbon, C. Robertson, E. Trucco, C. K. I. Williams, 2nd Edition, 2014, Wiley.
- https://scikit-learn.org/stable/
- https://matplotlib.org/3.3.1/contents.html
- https://www.geeksforgeeks.org/learning-model-building-scikit-learn-python-machine-learning- library/
- https://pypi.org/project/opencv-python/