Adaptive Background Mixture Models for Real-Time Motion Detection

Supervised by

Dr/ Alaa Hamdy

& Dr/ Rahma Ghouneim

Done by

Ahmed AbdelRahman 2021/03540

Bassel AbdelRahim 2021/07433

Introduction

This paper discusses the implementation and evaluation of Adaptive Background Mixture Models using the Gaussian Mixture Model (GMM) for real-time motion detection in videos. The primary goal is to detect real motion in videos with bimodal and multi-modal backgrounds under varying illumination conditions. The input for this project is a video, and the outputs are the foreground segmented objects and the background frame sequence. And we used two methods for this project, the GMM model and the GMM based background subtraction.

1.GMM-Based Background Subtraction:

1.1 Implementation

The implementation utilizes OpenCV to create and apply the GMM-based background subtractor. The GMM-based background subtractor handles videos with complex and multi-modal backgrounds, as well as varying illumination conditions, effectively. Morphological operations reduce noise in the foreground mask, resulting in clearer and more accurate segmentation of moving objects. The following steps outline the process:

1.1.1 Initialization:

- **Video Capture Object:** Initialize to read frames from the input video file.
- **GMM-based Background Subtractor:** Create using OpenCV's ‘`cv2.createBackgroundSubtractorMOG2`’.

1.1.2 Frame Processing Loop:

- **Reading Frames:** Frames are read sequentially from the video.
- **Foreground Mask Generation:** Apply the GMM-based background subtractor to each frame to generate a foreground mask.
- **Noise Removal:** Use morphological operations to remove noise from the foreground mask.
- **Foreground Image Extraction:** Extract the foreground image using a bitwise-AND operation with the foreground mask.

- **Background Image Extraction:** Retrieve the current background model image from the background subtractor.
- **Storage:** Store the foreground and background images in respective lists.
- **Visualization:** Display every 10 frames for visual inspection using matplotlib.
- **Frame Processing Indicator:** Print the index of each processed frame.

1.1.3 Termination:

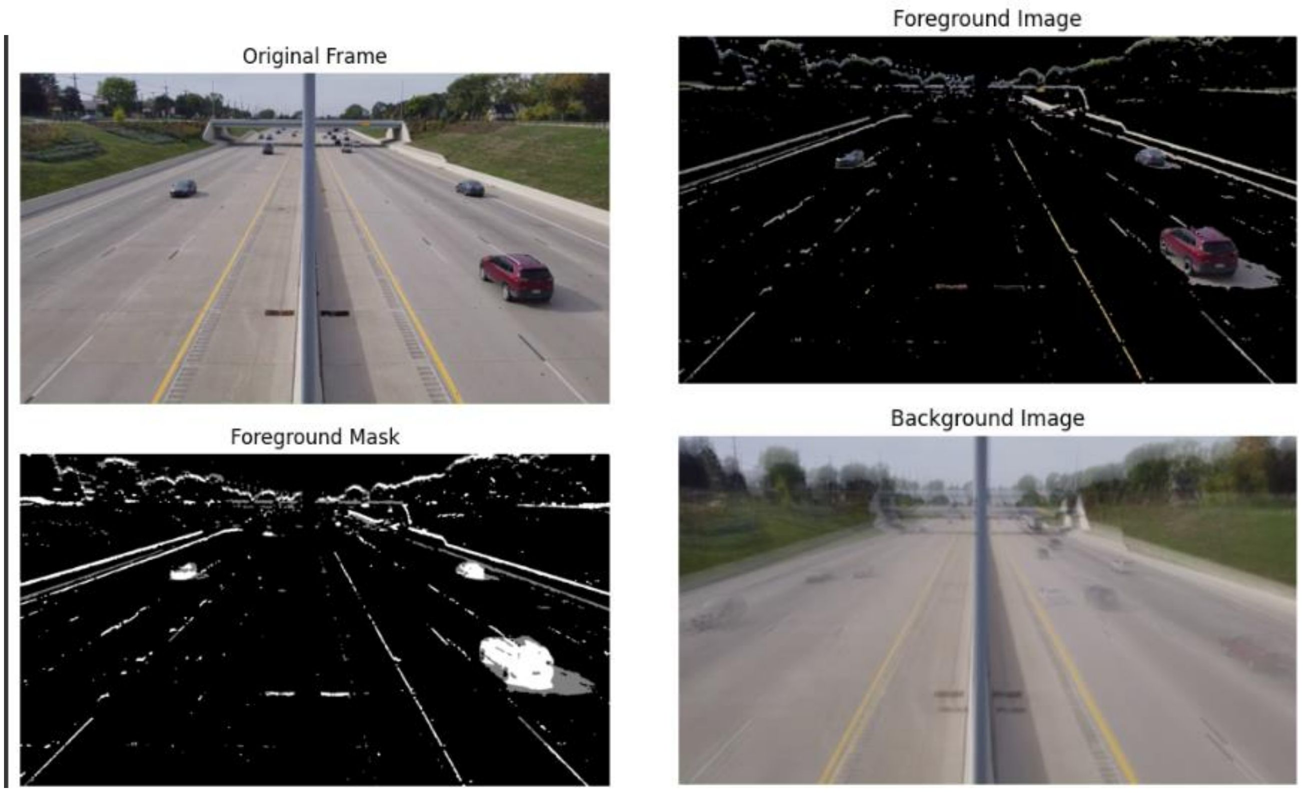
- **Loop Exit:** Manually exit the loop by pressing the 'q' key.
- **Resource Release:** Release the video capture object and close all OpenCV windows.

1.2 Results

The implemented code effectively detects moving objects in the video and segments them as foreground objects while updating the background model. The processed results are displayed every 10 frames, showing:

- **Original Frame:** The raw input frame from the video.
- **Foreground Mask:** A binary mask indicating detected foreground objects.
- **Foreground Image:** The segmented foreground objects superimposed on the original frame.
- **Background Image:** The estimated background model without the foreground objects.

-Some Snippets from the results:



2. GM Model:

2.1Implementation:

The project utilizes a custom implementation of the GMM to detect motion in video frames. The process involves the following steps:

2.1.1 Initialization:

- **Video Capture Object:** A video capture object is initialized to read frames from the input video file.
- **GMM Parameters:** Parameters for the GMM, such as the number of Gaussian components, the number of initializations, and the type of covariance parameters, are defined.

2.1.2 GMM Application:

- **Frame Reshaping:** Each video frame is reshaped to a 2D array where each row represents a pixel and its RGB values.
- **GMM Fitting:** The GMM is fitted to the reshaped frame to classify pixels into different Gaussian components.
- **Foreground Mask Creation:** The most common Gaussian component is assumed to represent the background, and a foreground mask is created by identifying pixels that do not belong to this component.
- **Noise Removal:** Morphological operations are applied to the foreground mask to remove noise.

2.1.3 Visualization and Storage:

- **Foreground Image Extraction:** The foreground image is extracted using the foreground mask.
- **Display:** The original frame, foreground mask, and foreground image are displayed using matplotlib for visual inspection.
- **Frame Processing Indicator:** The index of each processed frame is printed to the console.

2.1.4 Termination:

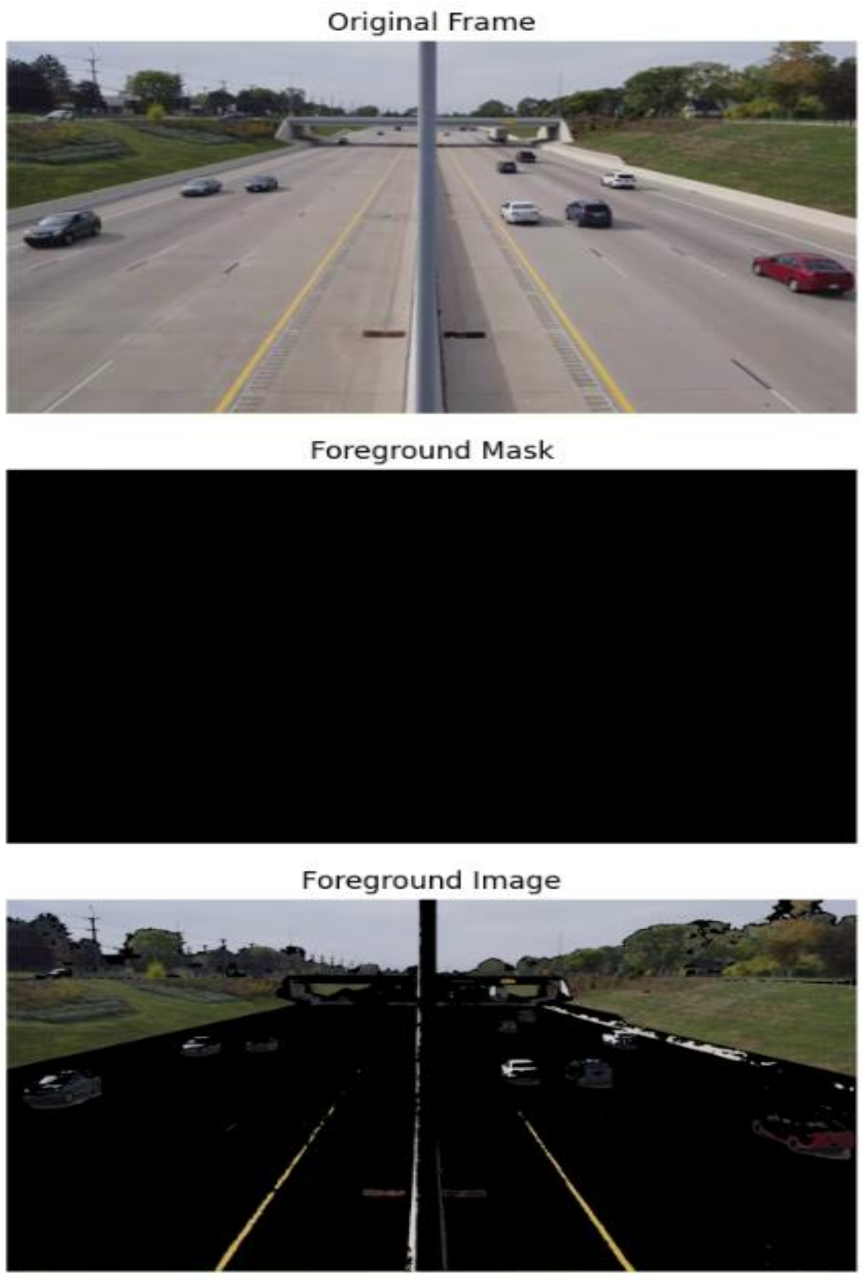
- **Loop Exit:** The loop can be exited manually by pressing the 'q' key.
- **Resource Release:** The video capture object is released, and all OpenCV windows are closed.

2.2 Results:

The implemented code successfully detects moving objects in the video and segments them as foreground objects while identifying the background. The processed results for each frame include:

- **Original Frame:** The raw input frame from the video.
- **Foreground Mask:** A binary mask indicating detected foreground objects.
- **Foreground Image:** The segmented foreground objects superimposed on the original frame.

-Some Snippets from the results:

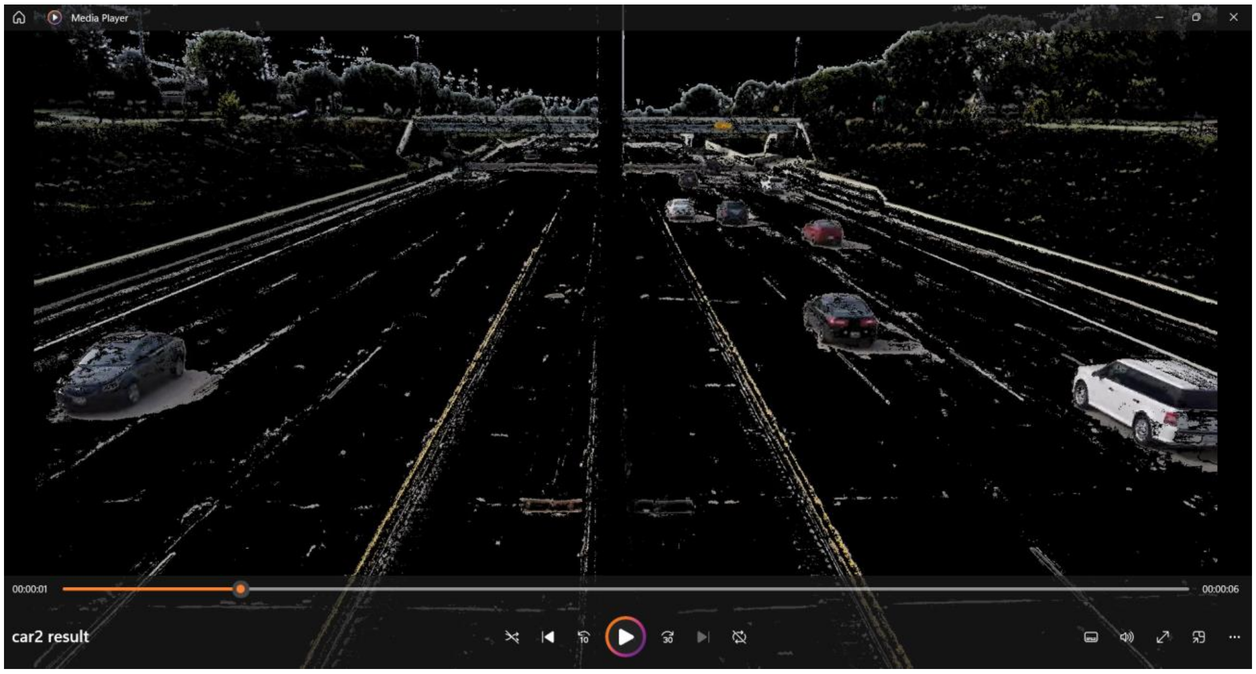


3. Additional Feature: Video Processing

In addition to the adaptive background mixture models for real-time motion detection, the project includes a feature for video processing and foreground segmentation. This section details the implementation of a video processing pipeline using OpenCV, which reads an input video, applies background subtraction to segment the foreground objects, and writes the processed frames to a new video file.

This feature enhances the project by providing a practical application of background subtraction for video processing. It enables the creation of a new video file that contains only the segmented foreground objects, which can be useful for various applications such as surveillance, traffic monitoring, and object tracking. The use of cv2.createBackgroundSubtractorMOG2 ensures robust performance in different lighting conditions and background complexities.

By integrating this additional feature, the project demonstrates a comprehensive approach to real-time motion detection and video processing using GMM-based background subtraction. This further supports the conclusion that GMM-based background subtraction is an effective and reliable model for handling diverse and challenging video processing tasks.



4. Conclusion:

The two methods presented an in-depth analysis of adaptive background mixture models for real-time motion detection using the Gaussian Mixture Model (GMM). The first method focused on the implementation and evaluation of a GMM-based background subtractor, leveraging OpenCV's cv2.createBackgroundSubtractorMOG2 to detect and segment foreground objects in videos with complex backgrounds and varying illumination conditions. The second method detailed a custom implementation of GMM, applying the model directly to video frames to achieve similar objectives.

In conclusion, the GMM-based background subtraction method proved to be the best model for real-time motion detection. Its ability to effectively segment foreground objects and adapt to varying background complexities and illumination conditions makes it a reliable and robust choice for practical applications in motion detection and video observation.

5. ACKNOWLEDGMENT:

We would like to express how grateful we are to Dr. Alaa Hamdy for being an amazing educator, and for wisely guiding and inspiring us to be better learners and high performing academics. And of course, we can’t forget to mention our heartfelt gratitude and honor to be part of the Misr International University student community, and for the amazing opportunity, learning environment, and beneficial resources provided for us throughout the years.