



# Mini Documentation — Background Subtraction with OpenCV



## Overview

This project uses **OpenCV's** `BackgroundSubtractorMOG2` to perform background subtraction in real-time. It can work with a **webcam feed** or a **video file**, creating a foreground mask that highlights motion.



## File Structure

```
background-subtraction/  
├── main.py          # Core logic for background subtraction  
├── vid1.mp4         # Optional video input (can be replaced)  
├── README.md        # Project overview and setup  
└── docs.md          # Mini documentation (this file)
```



## BackgroundSubtractorMOG2 Parameters

```
cv.createBackgroundSubtractorMOG2(history=20, varThreshold=50)
```

- **history (int):** Number of previous frames to build the background model.
  - Lower values make it more adaptive to changes.
- **varThreshold (int):** Threshold to determine whether a pixel is part of the background.
  - Lower value = more sensitive to motion.



## How It Works

### 1. Capture Video:

```
video = cv.VideoCapture('vid1.mp4') # or use 0 for webcam
```

## 2. Apply Subtractor to Each Frame:

```
mask = subtractor.apply(frame)
```

## 3. Display the Result:

```
cv.imshow('Mask', mask)
```

## 4. Exit Logic:

Press `'x'` or `'X'` to stop the video and close the window.

## 5. Video Looping:

Automatically reloads the video when it ends.

---

# Usage Instructions

## Run the script

```
python main.py
```

Make sure the video file `vid1.mp4` is in the same directory, or switch to webcam input by changing:

```
video = cv.VideoCapture(0)
```

## Output Example

- Original frame (not shown, but available for future enhancement)
- Foreground mask highlighting moving elements in white
- Static background appears black

---

## Applications

- Motion detection
  - Surveillance systems
  - Traffic monitoring
  - Human activity recognition
- 

## Tips & Extensions

- Want to show original + mask side by side?

```
combined = cv.hconcat([frame, cv.cvtColor(mask, cv.COLOR_GRAY2BGR)])  
cv.imshow('Original + Mask', combined)
```

- Want to save the mask output?

Use `cv.VideoWriter()` to store results as video.

---

## Dependencies

- Python 3.6+
  - OpenCV ( `pip install opencv-python` )
-