2024/06/14

# CV Final Project

## *Real-Time Monitoring of Door Status in Public Transit Systems*


喵喵喵喵

*R12941055* 鍾皓勳、*R12521503* 吳　悠

*R12521504* 吳玉文、*R12521522* 林佳駿

# OUTLINE

# OUTLINE

# Objective

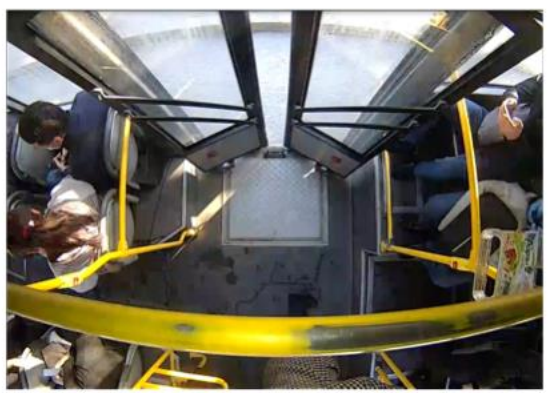
## Detect Door Status in Public Transit Systems

- *Using Data from Camera Video*
- *Limited Data*
- *Single door/ Double door*
- *Fisheye Lens*

Closed

Opening

Open



Closing

# Objective

**In order to detect the door status,**

**we need to overcome the following interferences…**

## *Interference*

- ***Variations in lighting conditions***
- ***Reflections from glass surfaces***
- ***Movement of the vehicle***
- ***Occlusions caused by passengers***

# OUTLINE

# Data Preprocess

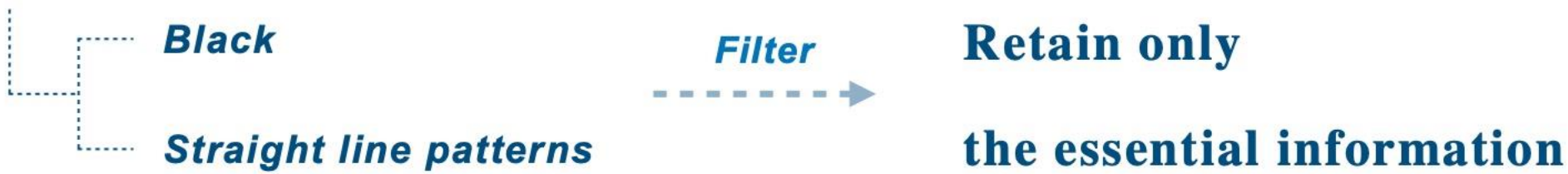
**In order to know the door status…**

↓

**Detect the door**

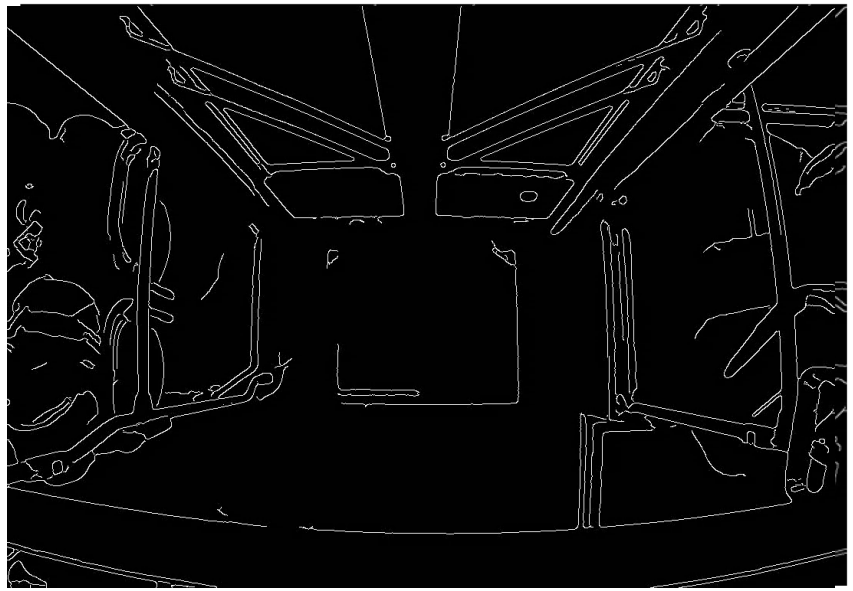
↓

**Observe the door first !**



# Data Preprocess

✓lighting conditions · ✓glass reflection · vehicle movement · passenger occlusions



## Canny - detect edge

- Gray frame
- Normalize gray frame
- Bilateral filter - reduce noise

- Contain many unnecessary Information

# Data Preprocess

✓ lighting conditions · ✓ glass reflection · ✓ vehicle movement · ✓ passenger occlusions

Retain door information
while filtering out noise

**Use masks to**
**retain the black portions**

**+ Canny - detect edge**
**+ HoughLinesP - detect straight lines**

Parameters:

- edges
- rho
- theta
- threshold
- minLineLength
- maxLineGap

# Data Preprocess

# Optical Flow

*The pattern of apparent motion of image objects*

*between two consecutive frames*

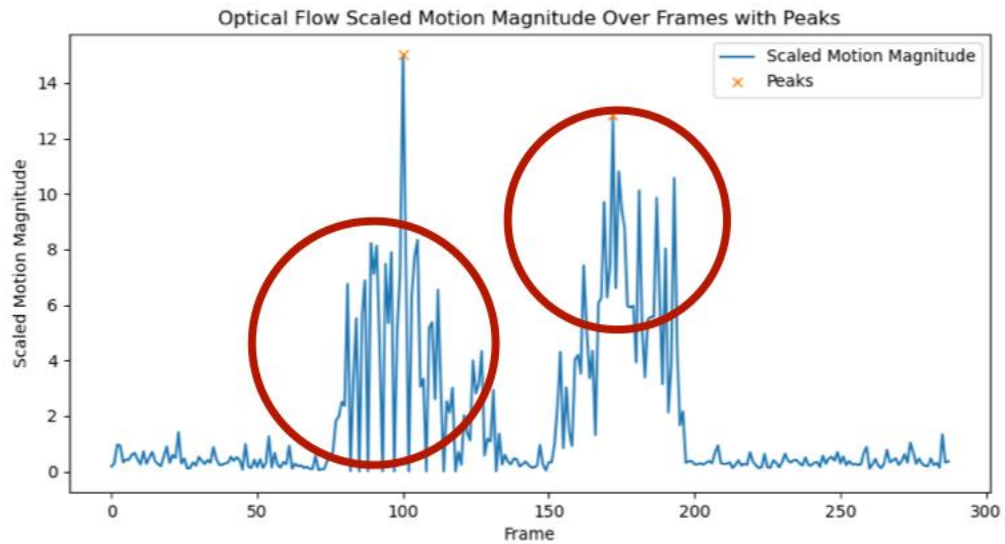
*caused by the movement of object or camera*

**Describe the movement direction and speed of each pixel**

**in the image from one frame to the next**

# Optical Flow

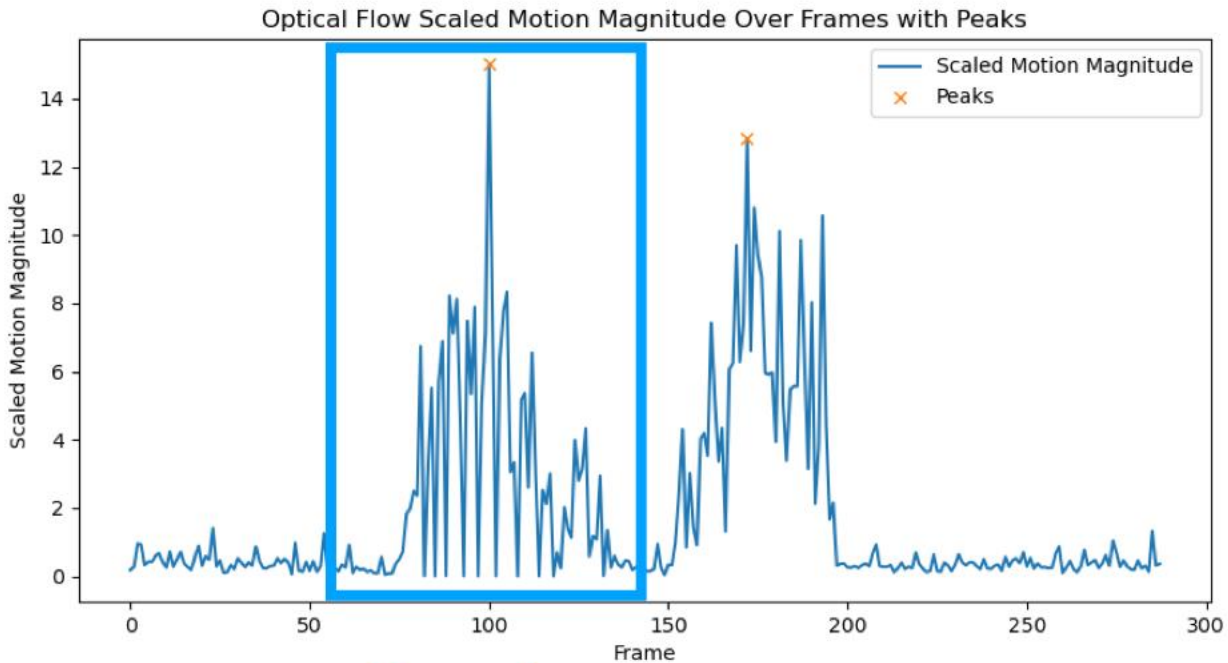
**Bus door is opening / closing**




***PEAKS !***

**the motion dynamics change significantly**

# Optical Flow




**Opening**

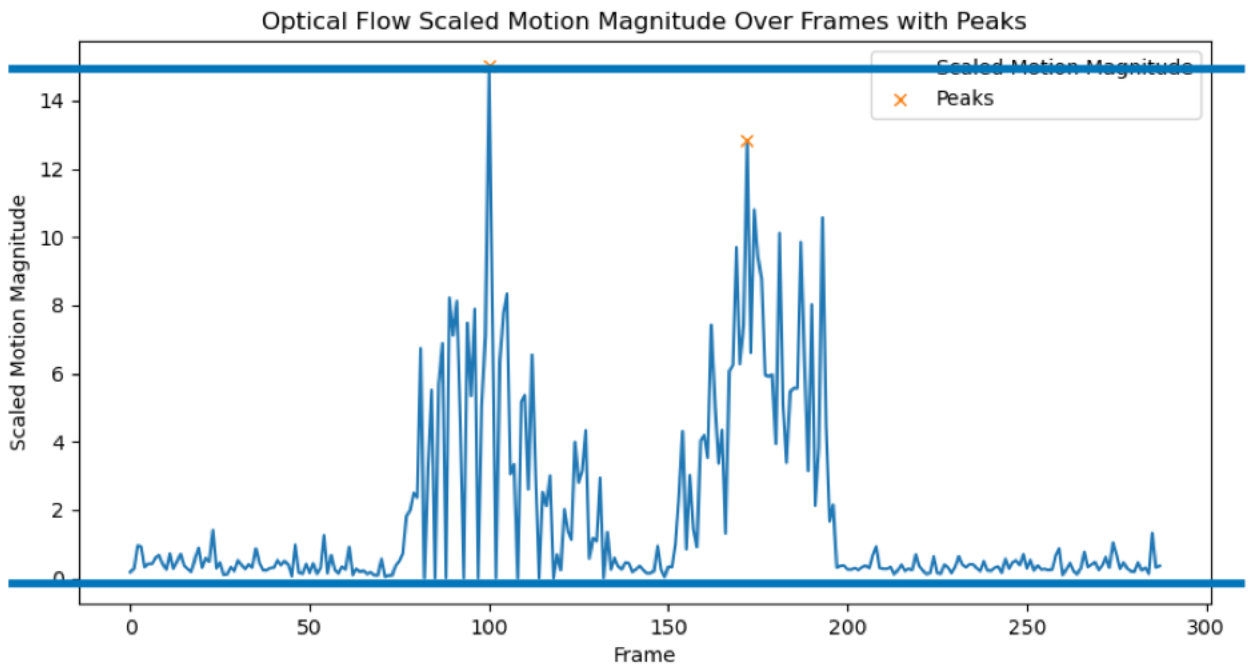
***Contain multiple peaks in one opening / closing status***

***In order to prevent catch two peak in one status***

***Merge close peak***

***See close peaks as one peek***

# Optical Flow



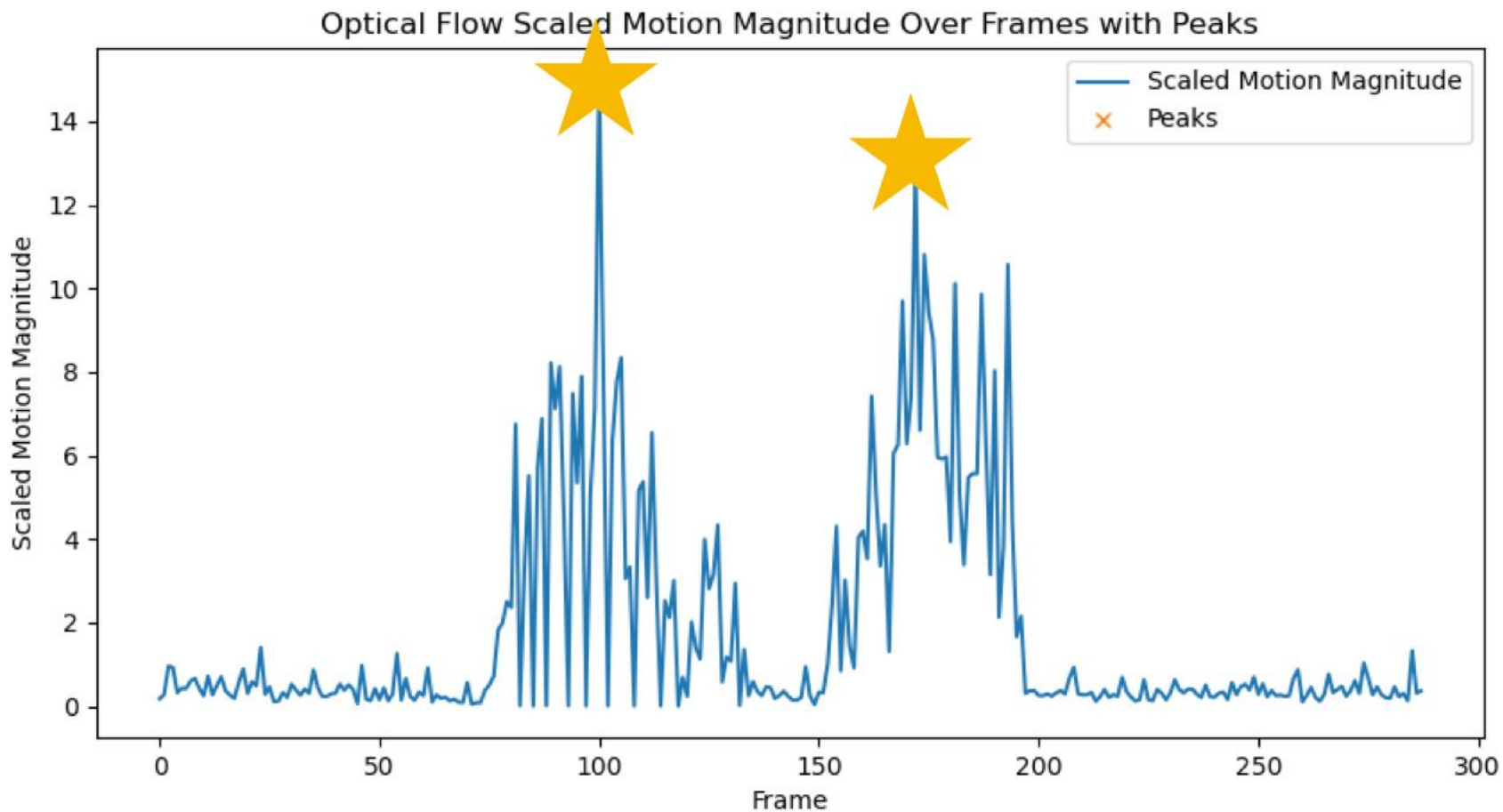

***Different video size will have different peak scale***

***In order to define an general threshold for peak detection***

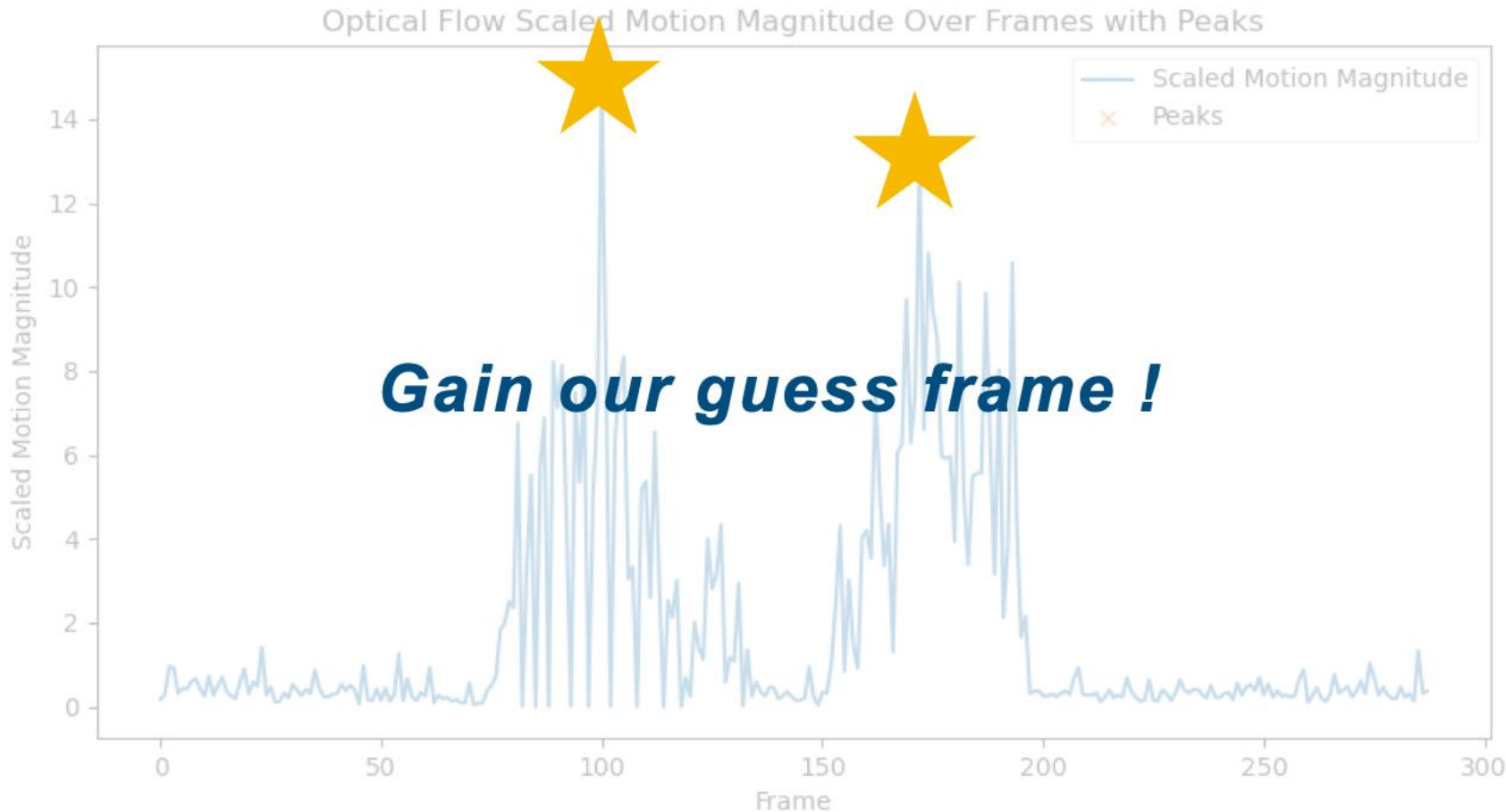
***Resize peak scale 0 - 15***

***Peak threshold = 7***

# Optical Flow



# Optical Flow




**Gain our guess frame !**

# Acceleration

## `ThreadPoolExecutor`

*An Executor subclass that uses a pool of threads to execute calls asynchronously*

↓

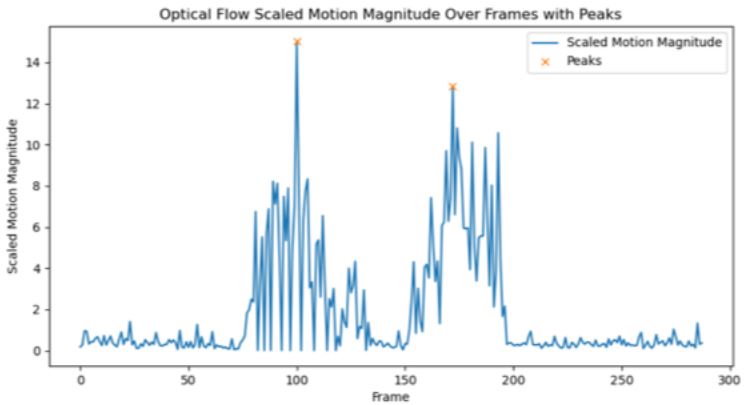
*6-7 min → < 2min*

**Approximately six times faster**

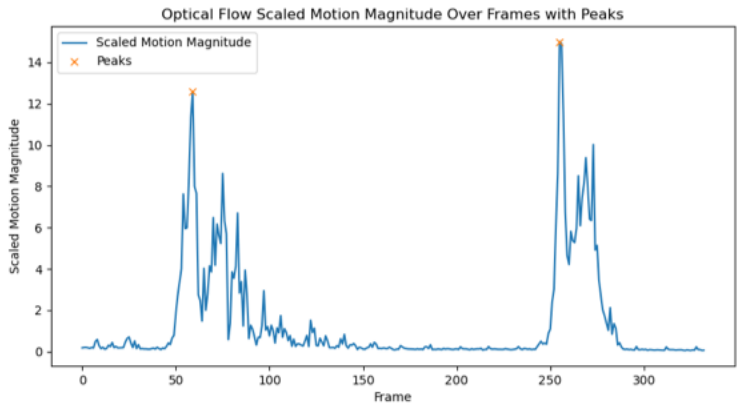
# OUTLINE

# Experimental Results

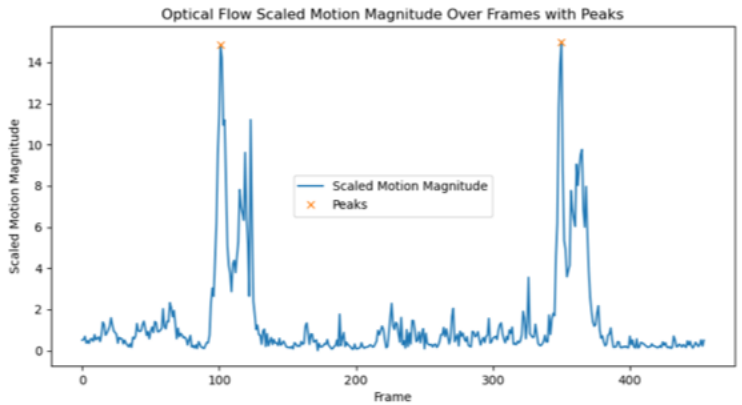
## Test 01



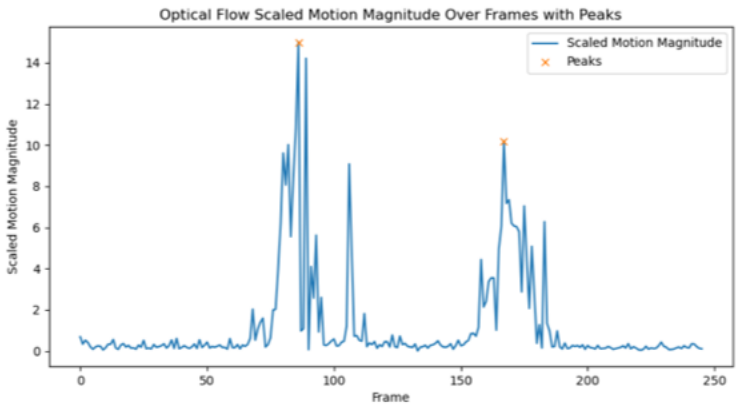

## Test 03



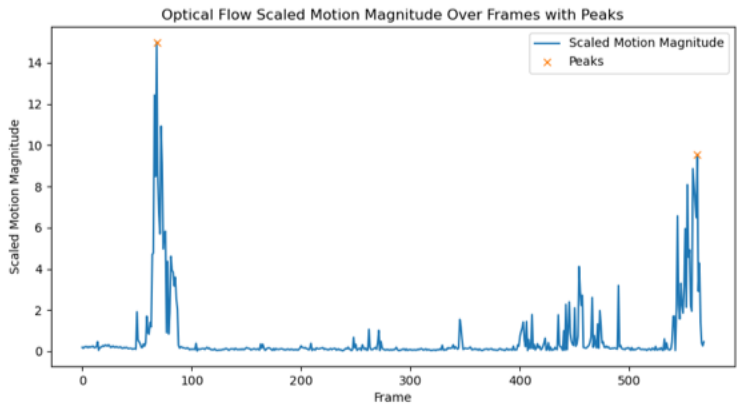

## Test 05




## Test 07

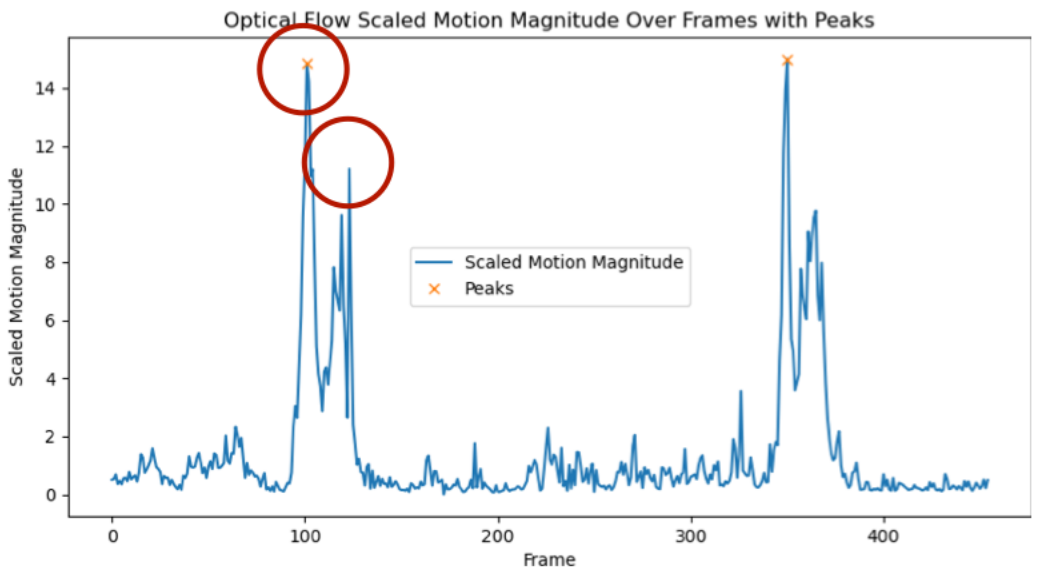



## Test 09

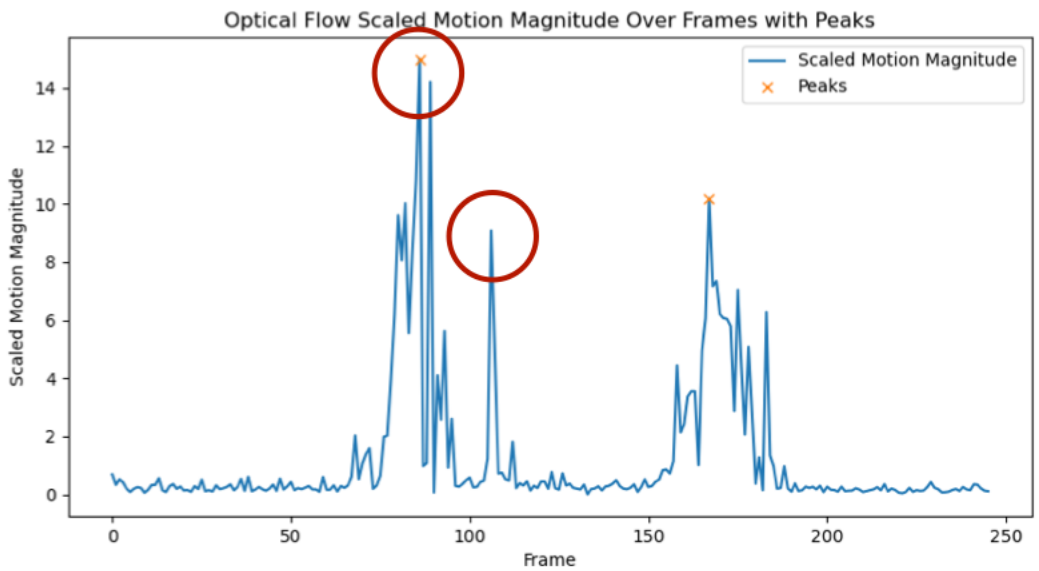


# Experimental Results

## Test 05




## Test 07



# Thank you for listening !

# Questions ?