MotionEstimationandEventDetectioninVideo

Objective

ThisprojectdetectsmotionandsignificanteventsinavideousingframedifferencingandhistogramcomparisonmethodsinOpenCV.Weidentifyregionsofmotionbyanalyzingchangesinconsecutiveframesandhighlightmomentswheresignificantmotioneventsoccurbasedonthe intensity of motion. n. n.

# Step1:LoadingandPreprocessingtheVideo

## ThefirststepinvolvesloadingthevideousingOpenCV.Eachframeofthevideoisresizedforeasiervisualization,andweconverteachframetograyscaletosimplifymotionanalysis.Grayscaleframesreducecomputationsinceweonlyworkwithonecolorchannel,whichisidealfor

detectingintensity-basedchanges.

import cv2

import numpy as np

import matplotlib.pyplot as plt

# Load the video

video\_path="/Users/AravindKandhasaamy/Documents/Work/Sem7/MachineVision/assignment4/zinger1.mp4" cap = cv2.VideoCapture(video\_path)

#Checkifvideoloadedsuccessfully if not cap.isOpened():

print("Error:Couldnotopenvideo.") exit()

# Parameters for motion detection

motion\_threshold=0.1#Adjustedthresholdbasedonobservedvalues event\_detected\_frames = []

motion\_intensity\_values = []

#Resizescalefactorforsmallerdisplay resize\_factor = 0.5

# Read the first frame

ret,previous\_frame=cap.read() if not ret:

print("Error:Couldnotreadthefirstframe.") exit()

# Step2:FrameDifferencingforMotionDetectionandThresholdingforEventDetection

## Todetectmotion,weuseframedifferencingtocompareconsecutiveframes.Bycalculatingtheabsolutedifferencebetweenconsecutivegrayscaleframes,weidentifyareaswherechangesoccur,whichhelpsusdetectregionsofmotion.Wethencalculateahistogramoftheframe difference to quantify motion intensity, which allows us to track how much change has occurred between frames.

Todetectsignificantmotionevents,wesetamotionintensitythreshold.Whenthecalculatedmotionintensityexceedsthisthreshold,itindicates that a significant change or event has occurred.These events are marked in the video frames for easy visualization.

# Resize and convert to grayscale

previous\_frame=cv2.resize(previous\_frame,(0,0),fx=resize\_factor,fy=resize\_factor) previous\_gray = cv2.cvtColor(previous\_frame, cv2.COLOR\_BGR2GRAY)

frame\_count=0 while True:

# Read the next frame

ret,current\_frame=cap.read() if not ret:

break frame\_count+=1

# Resize and convert to grayscale

current\_frame=cv2.resize(current\_frame,(0,0),fx=resize\_factor,fy=resize\_factor) current\_gray = cv2.cvtColor(current\_frame, cv2.COLOR\_BGR2GRAY)

# Compute frame difference

diff frame = cv2.absdiff(previous gray, current gray)

#Calculatemotionintensityusinghistogramcomparisonwithcorrelationmetric hist\_previous = cv2.calcHist([previous\_gray], [0], None, [256], [0, 256])

hist\_current = cv2.calcHist([current\_gray], [0], None, [256], [0, 256]) motion\_intensity=cv2.compareHist(hist\_previous,hist\_current,cv2.HISTCMP\_CORREL) motion\_intensity\_values.append(motion\_intensity)

# Threshold the difference image to get regions of motion

\_, motion\_mask = cv2.threshold(diff\_frame, 25, 255, cv2.THRESH\_BINARY)

# Detect significant motion events

ifmotion\_intensity<motion\_threshold:#Lowervaluesindicatesignificantdifferenceforcorrelation event\_detected\_frames.append((frame\_count, cap.get(cv2.CAP\_PROP\_POS\_MSEC)))

cv2.putText(current\_frame, f"Event Detected", (10, 30), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (0, 0, 255), 1)

# Highlight moving regions on the frame

contours,\_=cv2.findContours(motion\_mask,cv2.RETR\_EXTERNAL,cv2.CHAIN\_APPROX\_SIMPLE) for contour in contours:

ifcv2.contourArea(contour)>500:#Filteroutsmallcontours x, y, w, h = cv2.boundingRect(contour)

cv2.rectangle(current\_frame, (x, y), (x + w, y + h), (0, 255, 0), 1)

# Show the frame with highlighted motion cv2.imshow("MotionDetection(Resized)",current\_frame)

# Break the loop if 'q' is pressed

ifcv2.waitKey(10)&0xFF==ord('q'): break

# Update for the next frame previous\_gray = current\_gray

#Releasevideocaptureandclosewindows cap.release()

cv2.destroyAllWindows() cv2.waitKey(1)

#Printdetectedeventswithtimestamps print("Detected Events:")

for frame, timestamp in event\_detected\_frames:

print(f"Event detected at frame {frame}, timestamp: {timestamp:.2f} ms")

Detected Events:

# Step4:VisualizingMotionIntensityandDetectedEvents

## Themotionintensityisplottedovertheframestovisualizechangesinmotionthroughoutthevideo.Significantevents,wherethemotionintensityexceedsthethreshold,aremarkedwithreddotsonthegraph.Thisprovidesavisualrepresentationofwhenandwhereeventsoccur.

#Plotmotionintensityovertime plt.figure(figsize=(10, 6))

plt.plot(motion\_intensity\_values, label='Motion Intensity') plt.axhline(y=motion\_threshold,color='r',linestyle='--',label='MotionThreshold') plt.title("Motion Intensity Over Frames")

plt.xlabel("Frame") plt.ylabel("MotionIntensity") plt.legend()

# Highlight frames with detected events on the plot event\_frames=[frameforframe,\_inevent\_detected\_frames]

event\_intensity = [motion\_intensity\_values[frame - 1] for frame in event\_frames] plt.scatter(event\_frames,event\_intensity,color='red',marker='o',label='DetectedEvents')

plt.legend() plt.show()



