**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **1 june 2020** | **Name:** | **Shreya poojary** |
| **Course:** | **python** | **USN:** | **4al16ec074** |
| **Topic:** | **udemy** | **Semester & Section:** | **8-B** |
| **Github Repository:** | **Shreya-test** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| |  | | --- | |  | |  | |  | |  | | | **Report:** Adaptive Threshold By using this technique we can apply thresholding on small regions of the frame. So the collective value will be different for the whole frame.  # importing the necessary libraries  import cv2  import numpy as np  # Creating a VideoCapture object to read the video  cap = cv2.VideoCapture('sample.mp4')  # Loop untill the end of the video  while (cap.isOpened()):  # Capture frame-by-frame  ret, frame = cap.read()  frame = cv2.resize(frame, (540, 380), fx = 0, fy = 0, interpolation = cv2.INTER\_CUBIC)  # Display the resulting frame  cv2.imshow('Frame', frame)  # conversion of BGR to grayscale is necessary to apply this operation  gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)  # adaptive thresholding to use different threshold  # values on different regions of the frame.  Thresh = cv2.adaptiveThreshold(gray, 255, cv2.ADAPTIVE\_THRESH\_MEAN\_C,  cv2.THRESH\_BINARY\_INV, 11, 2)  cv2.imshow('Thresh', Thresh)  # define q as the exit button  if cv2.waitKey(25) & 0xFF == ord('q'):  break  # release the video capture object  cap.release()  # Closes all the windows currently opened.  cv2.destroyAllWindows() smoothing: Smoothing a video means removing the sharpness of the video and providing blurriness to the video. There are various methods for smoothing such as cv2.Gaussianblur(), cv2.medianBlur(), cv2.bilateralFilter().  For our purpose, we are going to use cv2.Gaussianblur().  # importing the necessary libraries  import cv2  import numpy as np  # Creating a VideoCapture object to read the video  cap = cv2.VideoCapture('sample.mp4')  # Loop untill the end of the video  while (cap.isOpened()):  # Capture frame-by-frame  ret, frame = cap.read()  frame = cv2.resize(frame, (540, 380), fx = 0, fy = 0,  interpolation = cv2.INTER\_CUBIC)  # Display the resulting frame  cv2.imshow('Frame', frame)  # using cv2.Gaussianblur() method to blur the video  # (5, 5) is the kernel size for blurring.  gaussianblur = cv2.GaussianBlur(frame, (5, 5), 0)  cv2.imshow('gblur', gaussianblur)  # define q as the exit button  if cv2.waitKey(25) & 0xFF == ord('q'):  break  # release the video capture object  cap.release()  # Closes all the windows currently opened.  cv2.destroyAllWindows() | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | |
|  |  |  |  | |
|  |  |  |  | |
|  | | | |
|  | | | |
|  | | | |