# **DPS920/CVI620 – Lab 1**

# **Introduction to OpenCV**

| Total Mark: | 10 marks (3% of the total course grade)   * 6 out of 10: Learn@Seneca submission (Due: Wednesday September 13th end of day) * 4 out of 10: Lab Demo (During Lab of Week 2) |
| --- | --- |
| Submission file(s): | * Lab01.docx (this document with your answers) * Team\_Contract.docx |

Name – Davender Singh

Student ID – 125700211

Please work in **groups** to complete this lab. This lab is worth 3% of the total course grade and will be evaluated through your written submission, as well as the lab demo. During the lab demo, group members are *randomly* selected to explain the submitted solution. Group members who do not present during the lab demo will lose the demo mark.

Please submit the submission file(s) through Learn@Seneca. ALL team members must submit the FINAL work.

***Please paste the resulting images and answers in this document.***

## **Part I: The Team Contract**

Make groups of **two** (or three) to work on the assignments for this course. Ideally, you will stick together for the rest of the term ☺. Prepare and agree on a team contract to minimize future disagreements. This contract must contain “Team Procedures”, “Team Expectations” and “Consequences”. Note that the course instructor can void this contract. See the following links for samples:

* Guidelines for writing team contracts- University of Arizona: <http://math.arizona.edu/~kerimar/Team%20Contract.doc>
* Team Contract- MIT:
* <http://web.mit.edu/6.005/www/fa15/projects/abcplayer/team-contract/#team_contract>

Submit a signed copy of the contract as Team\_Contract.docx. (Adding your names at the end of the digital document is sufficient. Scanned signatures are NOT needed.)

## **Part II: Software Setup**

1. Follow the instructions in attached file to set up your environment:

**Setting Up Computer Vision Course Environment.pdf**

* + If you are planning to use any other setup, please refer to <https://opencv.org/> , for example, for v.4.7.0, follow: [OpenCV: Introduction to OpenCV](https://docs.opencv.org/4.7.0/df/d65/tutorial_table_of_content_introduction.html)
  + For support, please refer to online resources.

## **Part III: A simple OpenCV project**

1. Open **Anaconda Navigator**, then on **Home** tab, choose the **socv** environment.
2. Launch **Jupyter Notebook** from socv environment**.**
3. Create a new python file, then copy and paste the following code in a code block.

|  |
| --- |
| import cv2 as cv  # Start a video capture, using device's camera  cap = cv.VideoCapture(0)  # Check if video file opened successfully  if (cap.isOpened() == False):      print("Error opening video stream or file")  frame\_width = int(cap.get(3))  frame\_height = int(cap.get(4))  print("Frame width: " , frame\_width)  print("Frame height: " , frame\_height)  # Read until video is completed  while(cap.isOpened()):      # Capture frame-by-frame      ret, frame = cap.read()      if ret == False:          break        # Display the frame      cv.imshow('frame',frame)      key = cv.waitKey(25)      # Press Q on keyboard to exit      if key & 0xFF == ord('q'):          break    # Release the video capture  cap.release()  # Close all the frames  cv.destroyAllWindows() |

1. Run the code. You should see your webcam’s video feed. Press ‘q’ to exit.
2. Change the parameter for cv.waitKey() from 25 to 40, or 1000. What happens?

**Ans 6:** Changing the parameter for cv.waitKey() from 25 to 40 affects the frame rate of the video display.

cv.waitKey(25) means the program waits for 25 milliseconds for a key event. This corresponds to roughly 40 frames per second (1000ms/25ms = 40fps).

cv.waitKey(40) corresponds to 25 frames per second.

1. Briefly explain what the code is doing.

**Ans 7:**

- The program starts by importing the OpenCV library using the alias `cv`.

- It then attempts to initiate video capture using the device's default camera (indicated by `0`).

- If the video stream can't be opened, an error message is printed.

- The width and height of the video frames are then obtained and printed.

- In the `while` loop, the program continuously captures and displays the video frames.

- If no frame can be read (e.g., at the end of a video file, though unlikely in this context as it's live feed), the loop will break.

- The current frame is shown in a window named 'frame'.

- The program waits for 25 milliseconds (or another value, as per the instructions) for the user to press a key.

- Afer that, If we press the 'q' key, the loop will break, effectively ending the video feed.

- Finally, the video capture is released, and all OpenCV windows are destroyed.

## **Part IV: Group work**

1. Complete this declaration by adding your names:

I, Davender Singh, declare that the attached assignment is our own work in accordance with the Seneca Academic Policy. We have not copied any part of this assignment, manually or electronically, from any other source including web sites, unless specified as references. We have not distributed our work to other students.