# **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 January 18th end of day) * 4 out of 10: Lab demo (Due: During Workshop of Week 2) |
| --- | --- |
| Submission file(s): | * Lab01.docx (this document with your answers) * Team\_Contract.docx |

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 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 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: **SettingUpAnacondaVirtualEnvironmentOpenCV.pdf**
   * If you are planning to use any other setup, please refer to <https://opencv.org/> , for example, for v.4.5.3, follow: <https://docs.opencv.org/4.5.3/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 **ocv** environment.
2. Launch **Jupyter Notebook** from ocv 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?

When the parameter is set to 25 ms the video feed shows every movement smoothly. When it’s set to 40 ms the difference is not clear. However, when the parameter is set to 1000 ms the video feed shows every movement laggy. So we can conclude that by increasing the parameter for cv.waitKey() we will see the movements in the video feed with delay.

1. Briefly explain what you do and don’t understand in the code here:

First, we import cv2 which is the module for opencv in python. “cap” is the “VideoCapture” object which receives a file name or device index(camera) as argument. If the video file couldn’t be opened an error message will be shown. Then we set the video frame and height and print them to the screen. The read function will return true if the next frame has been grabbed otherwise it returns false. Imshow() is used to display the frame which is grabbed from the read() function in a window. The waitKey() function will return –1 if no key is pressed and if a key is pressed it returns the pressed key’s code. “key” is a 32 bit integer and 0xFF is an 8 bit binary (11111111), therefore the result of “key & 0xFF” is the ASCII value of the pressed key. Ord(‘q’) returns the ASCII value of the character which is passed to it.

## **Part IV: Group work**

1. Complete this declaration by adding your names:

We, Sarvin Raeisi, Ian Henriquez-------- (mention your names), 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.

1. Specify what each member has done towards the completion of this work:

|  | Name | Task(s) |
| --- | --- | --- |
| 1 | Sarvin | q6, q7 |
| 2 | Ian | team contract |
| 3 |  |  |