# **DPS920/CVI620 – Lab 2**

# **Capturing Video**

| Total Mark: | 10 marks (3% of the total course grade)   * 6 out of 10: Learn@Seneca submission (Due: Wednesday September 20th end of day) * 4 out of 10: Lab Demo (During Lab of Week 3) |
| --- | --- |
| Submission file(s): | * Lab02.py / Lab02.ipynb * Lab02.docx (this document with your answers) |

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: A photo booth application**

For this lab, you need a webcam, or a digital camera connected and installed on your machine.

1. Open **Anaconda Navigator**, then on Home tab, choose **ocv/socv** environment.
2. Launch PyCharm, Visual Studio Code, or Jupyter Notebook, whichever environment you would like to use, from ocv/socv environment. Create **Lab02.py** or **Lab02.ipynb** file.
3. Write code to capture and show the video stream from your webcam (or camera).
4. Print out the width and height of the video capture. (Use cap.get())
5. Add code to:

* Take a snapshot whenever the ‘x’ key is pressed.
* Crop 30 pixels around the snapshot image.
* Then pad it with a constant border 50 pixels wide (any color you wish).
* Save the snapshot. Use image names such as ‘image1.jpg’, ‘image2.jpg’, etc., automatically incrementing the filename counter. Paste one of your snapshots here:



* Show this image in a new window for 1 second. Then automatically close it and return to the camera feed.
* Exit when ‘q’ or ESC key is pressed.

## **Part II: Basic calculations**

For this part, just write your answers in this file and submit.

1. Calculating fps:
   1. Calculate the fps implemented in the code.

the fps of the video capture depends on the source (like a video file or camera). If we're capturing video from a file, its inherent fps will dictate the rate of frame capture. If we're capturing from a camera, the default fps of the camera or any manually set fps will dictate it

* 1. How would you change the code to have an fps of 25 frames per second?

If we're capturing from a video file, we would be bound by the inherent fps of that video. However, if we're capturing from a camera, we can attempt to set the fps using:

cap.set(cv.CAP\_PROP\_FPS, 25)

1. Calculating Compression Ratio:

Find image01.jpg and look at its properties:

* + - What is the image resolution (dimensions)? Does this match the width and height output of question 5? Why is that?
    - The image resolution (dimensions) is 680x520 pixels. This does not match the width and height output of the video capture, which is 640x480 pixels. The discrepancy arises from the cropping and padding operations applied to the image. In theory, given the cropping of 30 pixels and a 50-pixel border, we would expect the image to be 680x520 pixels. However, the actual saved image dimensions match these calculated values. This confirms that the transformations applied in the code (cropping and padding) are directly responsible for the difference in dimensions from the original video frame to the saved image.

Cropping:

The image was cropped by 30 pixels on all sides. This operation reduces the image's dimensions.

New width after cropping:

640 − 2(30) = 580

New height after cropping:

480 − 2(30) = 420

Padding:

A constant border of 50 pixels wide was added to all sides of the cropped image. This operation increases the image's dimensions.

New width after adding border:

580 + 2(50) = 680

New height after adding border:

420 + 2(50) = 520

* + - What is the bit depth?

Bit depth: 24

* + - What is the file size in bytes?

2,88,714 bytes

* + - If this image was not compressed, what would the file size be?

10,60,800 bytes

* + - Calculate the compression ratio as the ratio between the uncompressed size and the compressed size.

Compression ratio : 3

## **Part III: Group work**

1. Add this declaration to your file:

We, Davender and Liliya, 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 | Davender Singh | Part II – Basic calculations |
| 2 | Liliya Panfilova | Part I – A photo booth application |
| 3 |  |  |