# **Assignment 2 (10%):**

**Image Processing and Computer Vision**

*Using OpenCV*

DUE DATE: Friday, Feb 23, 2024, 11:59 PM ET

**Objectives:**

* Develop proficiency in computer vision techniques for image processing, object detection, and feature tracking.
* Gain hands-on experience with OpenCV, learning to apply various computer vision techniques.

**Assignment Guidelines:**

* All students in this course must read and meet the expectations described in the [Student Academic Integrity](https://intranet.laurentian.ca/policies/2017.09.19%20-%20Policy%20and%20Procedures%20on%20Academic%20Integrity%20-%20EN.pdf).
* Assignments must follow the programming standards document published on the course website in the D2L. Marks will be taken off if standards are not followed.
* **Submit just one (preferable) .py file AND the associate report in PDF per group**. Name the file based on your group “ID” and the assignment number, exactly as in this example for **assignment x and** **group x**: **CPSC\_5616EL\_Ax\_Gx.py**. Same naming convention applies to the PDF, **CPSC\_5616EL\_Ax\_Gx.pdf**.
* **Do NOT zip the files** that you submit.
* You may submit the assignment multiple times, but only the most recent version will be marked.
* After the due date and time, a late penalty of 2% per hour, or a portion of an hour, will be applied. After 49 hours, the penalty is 100% and submissions will no longer be accepted. The date and time of the last file submitted control the mark for the entire assignment.
* We compare all submitted assignments with one another, and pursue academic dishonesty vigorously. **You must complete the Honesty Declaration in the D2L before you will be able to submit your assignment.**

1. **Introduction**

In this assignmnet, you will utilise OpenCV 4.x, a widely-used library for computer vision, to create a module that facilitates the capture and track of objects from a live streaming webcam. This module will be an ongoing element throughout our series of assignments. Your solutions to these open-ended problems will be evaluated based on functionality, efficiency, coding style, and the elegance of your implementation.

**Project Focus**: Your task is to implement a [feature detection](https://docs.opencv.org/4.9.0/db/d27/tutorial_py_table_of_contents_feature2d.html) system capable of accurately identifying feature points and recognizing objects such as a ball and marker, specifically for the [FIRA Sport HuroCup](https://firaworldcup.org/leagues/fira-sports/hurocup/) competition. Refer to the game rules, for example, the [Marathon event](https://docs.google.com/document/d/1mC2gLOjVYGabGnAS96kaOjltj17cigHKygP_nKmUC_M/edit), for guidance.

1. **Setup and Image Preparation:**

* Install OpenCV 4.x and PyCharm on your machine (Linux preferred, but if you have problem with the webcam due to the VM, Windows and MacOS are permitted in this Assignment).
* Download the provided sample Point-of-View (POV) images, taken from approximately 30 cm away from the camera, available [[here](https://drive.google.com/drive/folders/1Hj3rfQmndPfRUW_qkSt6p0A5sgP-bqjX?usp=sharing)].
* Preprocess the streaming input image from the camera: resize to 640x480 (or 320x240 for faster processing), convert to various color spaces (RGB, HSV, grayscale), apply blurring, and perform edge detection. Each process should correspond to a specific OpenCV method (e.g., using Canny() for the Canny edge detector). Refer to the OpenCV documentation at [[http://docs.opencv.org/](https://docs.opencv.org/4.x/)].

1. **Ball Tracking (7%):**

* Develop a function trackBall() to track a ball (an orange/yellow tennis ball, approximately 6.5 cm in diameter) in the field. This function should determine the ball's 2D position and size from the camera's POV, as demonstrated in Figure 1.
* Utilise image processing techniques such as color extraction, thresholding, shape recognition, and morphological operations. Implement at least one method, for instance, ApproxPolyDP, Hough Circle Transform or [**Cascade Classifier**](https://docs.opencv.org/4.x/db/d28/tutorial_cascade_classifier.html) for contour detection.

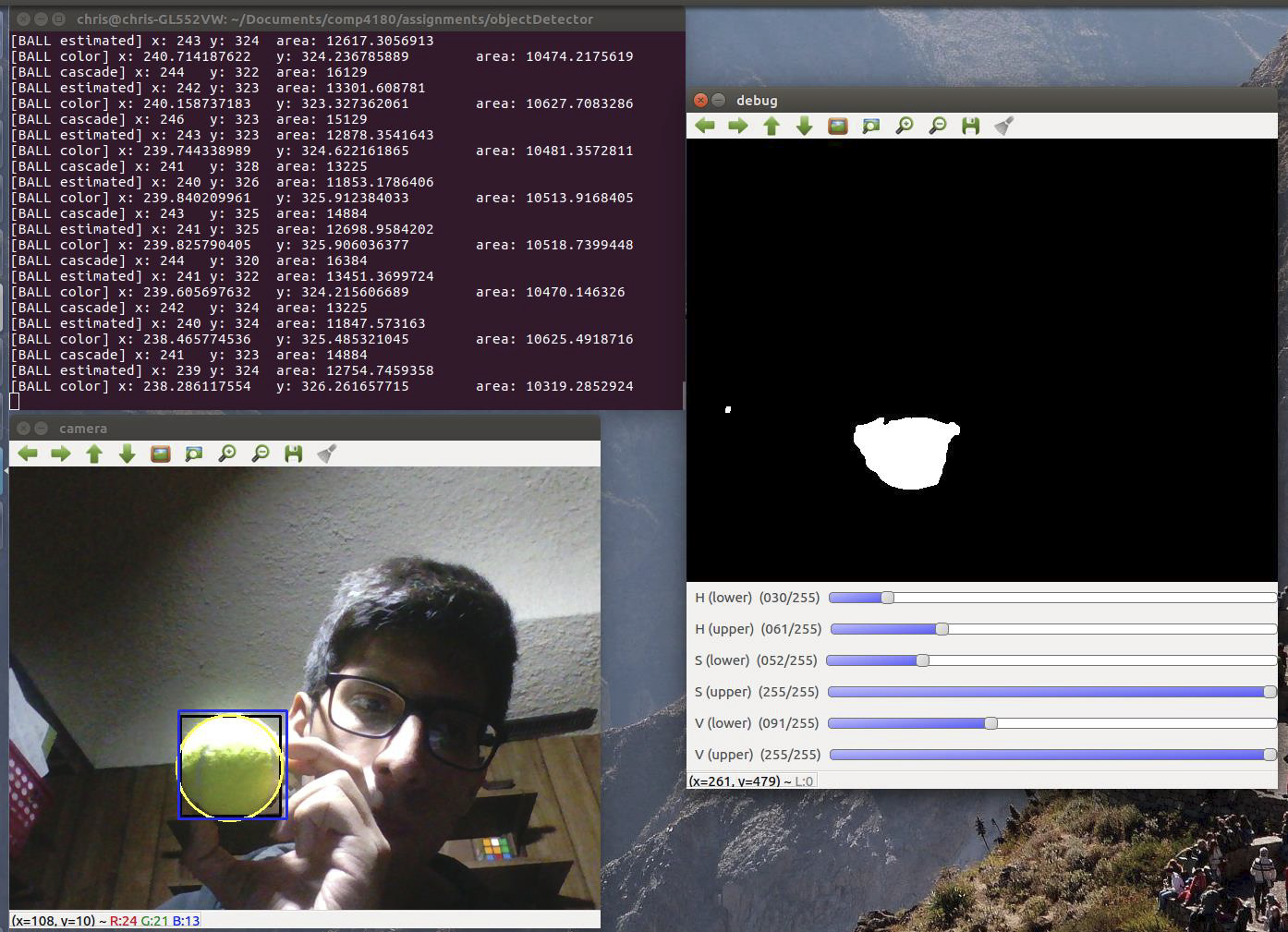


Figure 1

1. **Arrow Marker Tracking (8%):**

* Create a function trackArrows() to identify and extract the Region of Interest (ROI) for different 10 cm x 10 cm arrow patterns (**Left**, **Right**, and **Forward** as described in the [Marathon event](https://docs.google.com/document/d/1mC2gLOjVYGabGnAS96kaOjltj17cigHKygP_nKmUC_M/edit)) from the video stream.
* Employ techniques like shape recognition, ROI extraction, and perspective transformation for marker region extraction. Analyze the marker content to recognise different arrows as described in the game rules as shown in Figure 2.
* Implement at least one method of your choice, such as a feature detector, corner detector, template matching, geometric approach, or machine learning approach.



Figure 2

1. **Reporting (5%):**

* Write a 2-page report in Arial, 11-point font, single-spaced, detailing your implementation and results for tasks 3 and 4.
* Discuss the efficiency and accuracy of your outputs, including processing time and repeatability tests. **Incorporate examples of your image processing outputs as an image pipeline in your report.**
* Record a video demonstration (maximum 5 minutes) to showcase the real-time performance of your computer vision programming. You can upload it to YouTube and attach the link in the report.
* **Note**: Ensure that your code is well-commented to receive full marks for the assignment.

**Submission Guidelines:**

1. Submit your code as a **.py** file.
2. You may submit more than one .py file for different algorithms. Make sure you rename the filename accordingly.
3. Submit your report as a **PDF** file, ensure that all figures, and tables are properly labeled.
4. Your assignment should be self-contained, meaning a person should be able to understand your process and results just by reading your report and going through your code.

**Evaluation Criteria:**

1. Correctness of implementation.
2. Efficiency of the algorithms.
3. Quality and clarity of the code and report.
4. Depth of analysis in comparing the algorithms.

**Note on Group Contributions and Grading:**

If any group member believes that another member of their group deserves a lower grade due to their contribution level, they are encouraged to address this concern. To formalize this, the group can include an additional section in their report detailing the situation and the proposed grade adjustment, with the consent of all group members. It's essential that all group members agree and provide their consent for any proposed grade changes.

It's always best to communicate openly within your group and seek collaborative solutions. However, if discrepancies in contributions are significant and consensus is achieved, this mechanism ensures fairness in grading.

**Best of luck! Remember, the process and learning are as important as the final results.**