

## Part 3 – Coding on PC (out of 7 points)

The goal of this part is to detect the **ball** from the **Balle.zip Dataset** available on the course website: [https://melodiedaniel.github.io/deep\\_learning/](https://melodiedaniel.github.io/deep_learning/). All images in the **Dataset** contain a single ball that must be detected. Since the ball is relatively small, it is not necessary to define its position with a **bounding box**; a vector (x, y) representing the **position** of the **ball center** is sufficient. To comply with the **YOLO** label format, we have defined both **height** and **width** of the box as 0.05.

**Remember to submit your code on Moodle.**

**Note:** you will need to adapt the absolute path to the folder containing the Dataset in the **.yaml** file.

**Question 3.1.** (1.5 pt) Use **YOLO** for detection:

- Train with **YOLOv11**.
- Evaluate the training using the **test set**.

**Question 3.2.** (1 pt) Create the **three Datasets** for **train**, **validation**, and **test** using a **DataLoader**.

**Question 3.3.** (2 pt) Propose a **custom CNN** network called **BallDetectorCNN** for ball detection:

- Implement the **BallDetectorCNN** class to predict the (x, y) **position** of the ball center.
- Implement the training loop.
- Consider stopping the training if the validation loss increases.
- Calculate the percentage of correct detections on the test set.

**Question 3.4.** (0.5 pt) Compare the results of **YOLO** and the **custom CNN**. What do you observe in terms of **performance** and **training time**?

**Question 3.5.** (0.5 pt) Explain the difference in the **network output** between performing **classification**, **detection**, or **tracking** of the ball. Describe in 2–3 sentences what would need to be **modified** in the code.

**Question 3.6.** (1.5 pt) Propose an implementation of an **extension** of **BallDetectorCNN** to handle cases where some images do not contain a **ball**. For this question, training is not required.