**Use of yolo final project ai programming**

Table des matières

[Output during yolo the training of yolo model : 1](#_Toc185142678)

[Vocabulary: 1](#_Toc185142679)

[box\_loss (Perte des boîtes englobantes) 2](#_Toc185142680)

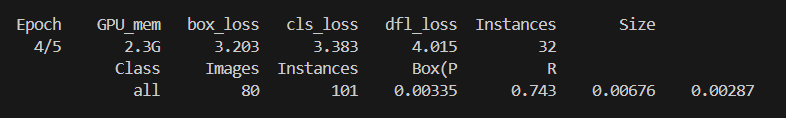
[dfl\_loss (Perte associée au Distance Focal Loss) 2](#_Toc185142681)

[Instance : 3](#_Toc185142682)

[Box(P, R, mAP50, mAP50-95): 3](#_Toc185142683)

[F1 Score 4](#_Toc185142684)

[Reuse of a trained model : 4](#_Toc185142685)

Output during yolo the training of yolo model :   


## Vocabulary:

Un **object**: est un objet annoté dans vos images, c'est-à-dire un objet présent dans les annotations de votre dataset. We call this the *ground truth.*

Une **instance :** fait référence à une **apparition individuelle** d’un objet dans une image donnée, qu'elle soit détectée correctement ou non.

A **threshold** is a value that determines the confidence level at which a detection is accepted.

For example:

* **Threshold = 0.5:** Only predictions with confidence ≥ 50% are accepted.

Explication :

| **Nom** | **Indique quoi ?** | **Bonne direction ?** |
| --- | --- | --- |
| GPU\_mem | Mémoire utilisée pour l'entraînement | Stable ou légèrement croissante |
| box\_loss | Précision de la localisation des objets | Doit diminuer au fil des époques |
| cls\_loss | Qualité de la classification des objets | Doit diminuer au fil des époques |
| dfl\_loss | Précision des coordonnées des boîtes englobantes | Doit diminuer au fil des époques |

### box\_loss (Perte des boîtes englobantes)

* **Description :** La **box loss** quantifie l'erreur entre les boîtes prédites par le modèle et les boîtes réelles annotées dans vos données.
* **Comment est-elle calculée :**
  + Utilise des mesures comme **CIoU (Complete Intersection over Union)** ou **DIoU** pour évaluer la qualité de la localisation des boîtes prédite.
  + Plus la boîte prédite est éloignée de la boîte réelle, plus cette perte est élevée.
* **Interprétation :**
  + Une **valeur élevée** (ex. 3.168 au début) indique que le modèle a encore beaucoup de difficulté à positionner correctement les boîtes englobantes.
  + Une **valeur qui diminue** au fil des époques montre que le modèle améliore la localisation des objets.

Erreur faible : 0.01 à 0.2 :

Erreur élevée : > 1

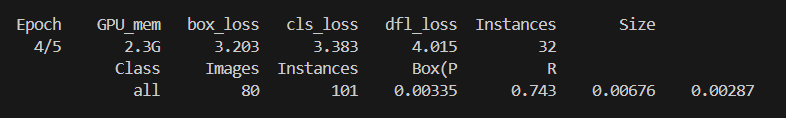
### dfl\_loss (Perte associée au Distance Focal Loss)

* **Description :** La **DFL loss** (Distance Focal Loss) améliore la précision des prédictions de positionnement des boîtes en affinant les coordonnées des boîtes englobantes.
* **Comment est-elle calculée :**
  + Basée sur une technique qui "pondère" les prédictions des coordonnées pour mieux ajuster les boîtes.
  + Cela aide le modèle à produire des boîtes bien centrées sur les objets détectés.
* **Interprétation :**
  + Une **valeur élevée** initiale (ex. 4.288) signifie que le modèle ajuste encore mal les bords des boîtes.
  + Une **baisse progressive** montre que les prédictions deviennent plus précises.

### Instance :

**L’Instance** totalise toutes les détections (correctes et incorrectes) sur l'ensemble des images. Si le modèle détecte 20 objets dans une seule image et rien ailleurs, cela comptera 20 instances.

### Box(P, R, mAP50, mAP50-95):



#### P (Precision):

* **P (Precision)**:  Precision quantifies the proportion of true positives among all positive predictions, assessing the model's capability to avoid false positives
* **Low Precision:** The model may be detecting too many non-existent objects. Adjusting confidence thresholds might reduce this.

#### Recall :

Recall calculates the proportion of true positives among all actual positives, measuring the model's ability to detect all instances of a class.

Le **recall** mesure la proportion d'objets annotés qui ont été détectés, **indépendamment de leur précision** ou de leur emplacement exact.

* **Rappel élevé (0.743)** signifie que votre modèle détecte **la majorité des objets présents** dans les images, même si :
  + Les détections sont mal localisées (faible IoU avec les annotations réelles).
  + Les classes sont parfois incorrectes.
* Cela reflète une capacité brute du modèle à **repérer les zones contenant des objets**, sans garantir que les prédictions soient précises ou fiables.

Recall = Détections  ​ / Objets annotées

Objets annotées = les objets annotées de notre data set de training.

#### mAP50:

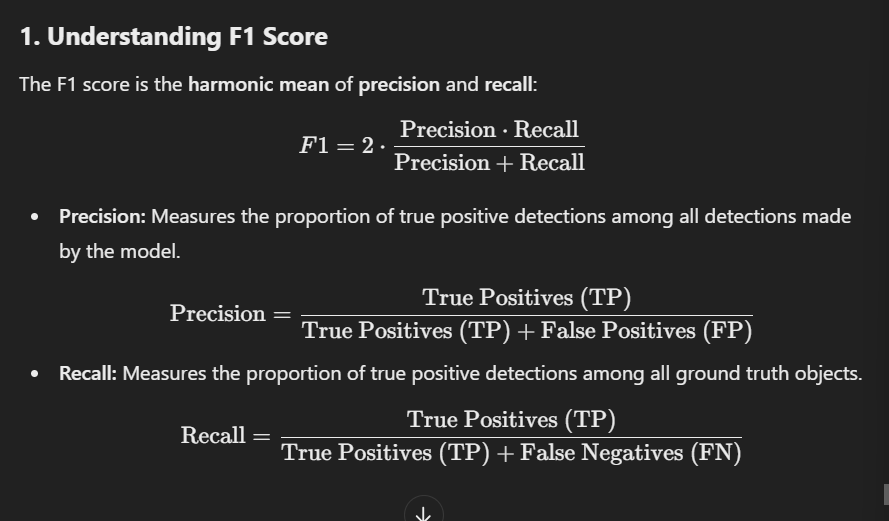
Mean average precision calculated at an intersection over union (IoU) threshold of 0.50. It's a measure of the model's accuracy considering only the "easy" detections.

#### mAP50-95:

The average of the mean average precision calculated at varying IoU thresholds, ranging from 0.50 to 0.95. It gives a comprehensive view of the model's performance across different levels of detection difficulty.

### F1 Score

* **F1 Score:** The F1 Score is the harmonic mean of precision and recall, providing a balanced assessment of a model's performance while considering both false positives and false negatives.



#### F1-confidence-curve :

The curve shows how the model's balance between precision and recall changes as you vary the threshold.

# Reuse of a trained model :

YOLOv8 saves the following files in the weights/ subfolder:

* **best.pt**: The best-performing weights during training (based on validation metrics like mAP).
* **last.pt**: The weights after the final epoch of training.

These are PyTorch format files and contain the trained parameters of your model.

To reuse the trained model, you can load the best.pt file like this:

from ultralytics import YOLO

# Load the trained model

model = YOLO("path/to/best.pt")

# Use the model for inference or further training

results = model("path/to/test/image.jpg") # Perform inference on a test image