UNIVERSITY ECHAHID HAMA LAKHDAR, EL-OUED
INSTITUTE OF EXACT SCIENCES
DEPARTMENT OF COMPUTER SCIENCE
2^{ed}*Master* :
ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
Semester : 3.2025

Lab2 Image Understanding and Convolutional Neural Networks

Master Level – Computer Vision

1. Context

Traditional object detectors such as **R-CNN**, **Fast/Faster R-CNN**, **SSD**, and **YOLO** rely on rectangular bounding boxes to localize objects. However, bounding boxes poorly represent circular, rotated, or irregular objects. In this assignment, you will go **beyond bounding boxes** by replacing them with **geometric and parametric representations** (circles, ellipses, and curves) and finally attempt to **improve FPS or accuracy** using the same dataset.

2. Objectives

— Compare the main object detection architectures : R-CNN, Fast/Faster R-CNN, SSD, and YOLO.
— Implement geometric and parametric object representations.
— Evaluate their effects on localization accuracy (IoU).
— Optimize your chosen detector for higher FPS or accuracy.

3. Dataset

Use one of the following :

— **Pascal VOC 2012**
— **COCO subset** (e.g., people, cars, dogs)
— A **custom dataset** with at least 100 labeled images.

4. Assignment Structure

Part A — Understanding the Detectors (20%)

1. Write a **comparative summary** of :
   nosep R-CNN, Fast R-CNN, Faster R-CNN, SSD, YOLO
   Include for each :
   nosep Backbone network
   nosep Region proposal method
   nosep Anchor mechanism
   nosep Loss function
   nosep Inference speed (FPS)
2. Run at least one pretrained detector (Faster R-CNN, SSD, or YOLOv8) and visualize the bounding boxes.

Part B — Bounding Circle Representation (20%)

- Convert each detected box $(x_{min}, y_{min}, x_{max}, y_{max})$ into a circle :

$$(x_c, y_c) = \left(\frac{x_{min} + x_{max}}{2}, \frac{y_{min} + y_{max}}{2}\right), \quad r = \frac{\sqrt{(x_{max} - x_{min})^2 + (y_{max} - y_{min})^2}}{2}.$$

- Draw circles using `OpenCV`.
- Compute the Intersection-over-Union (IoU) between each circle and ground-truth box.
- Discuss trade-offs versus rectangular bounding boxes.

Part C — Elliptic Representation (25%)

- Fit an ellipse around each object using :

$$(x_c, y_c, a, b, \theta)$$

  where a, b are semi-axes and θ the rotation angle.
- You may use :

```
cv2.fitEllipse(contour)
```

- Visualize ellipses and compute IoU(ellipse, ground truth).
- Compare accuracy and visual fit against circles and boxes.

Part D — Parametric Representation (25%)

- Represent object boundaries parametrically :

$$x(t) = x_c + a\cos(t) + \alpha\cos(2t), \quad y(t) = y_c + b\sin(t) + \beta\sin(2t)$$

  or fit B-spline / Bézier curves to contour points.
- Fit parameters via least-squares or optimization (e.g., `scipy.optimize`).
- Visualize parametric curves and compare IoU results.

Part E — Dataset-Driven Performance Enhancement (10%)

Use your current dataset to **increase either FPS or accuracy**. Possible directions include :

label=(a) Reducing or increasing input resolution to test the speed-accuracy tradeoff.

lbbel=(b) Re-clustering anchor boxes or ellipse priors via K-means.

lcbel=(c) Applying data augmentations (rotation, mosaic, blur, brightness).

ldbel=(d) Model pruning or quantization to improve FPS.

lebel=(e) Swapping heavy backbones (e.g., ResNet50) for lightweight ones (MobileNetV3).

lfbel=(f) Trying advanced losses (CIoU, EIoU) or Soft-NMS for localization refinement.

Deliverables for Part E :

- A table comparing baseline and improved performance :

Metric	**Baseline**	**Improved**	**Δ**
FPS			
mAP/IoU			

- A short discussion of your modification and results.

5. Deliverables

1. Jupyter Notebooks or Python scripts :
   nosep `PartA_Models.ipynb`
   nosep `PartB_Circle.ipynb`
   nosep `PartC_Ellipse.ipynb`
   nosep `PartD_Parametric.ipynb`
2. A 2–3 page PDF report containing :
   nosep Theoretical comparison table
   nosep Key equations (geometric and parametric)
   nosep Visual and quantitative results
   nosep Discussion of results and reflections

6. Grading Rubric

Section	**Weight**
Part A — Detector Analysis	20%
Part B — Circle Representation	20%
Part C — Ellipse Representation	25%
Part D — Parametric Curve	25%
Part E — FPS/Accuracy Improvement	10%
Bonus : End-to-end ellipse regression or differentiable fitting	+10%

7. Recommended Libraries

`OpenCV`, `NumPy`, `Matplotlib`, `SciPy`, `torchvision`, `ultralytics`, `skimage`, `onnxruntime`, `torch.quantization`

8. Suggested Timeline

Week	**Task**
1	Baseline model study (R-CNN/YOLO) and detections
2	Circle and ellipse implementation
3	Parametric curve fitting and visualization
4	Performance optimization and report writing

9. Expected Output Examples

— Visual comparison : box, circle, ellipse, and curve overlays.
— IoU table : comparing each representation.
— Before/after FPS and mAP results.

10. Optional Research Extension

For students interested in deeper exploration :
— Modify YOLO or Faster R-CNN to directly regress ellipse parameters.
— Evaluate differentiable geometric losses (EIoU, GIoU) for ellipse fitting.
— Apply to specialized domains (e.g., medical imaging, aerial object detection).

End of Assignment