

**Đề thi:**

**DEEP LEARNING WITH PYTHON / COMPUTER VISION WITH  
DEEP LEARNING**

**Thời hạn làm bài: 00h00 ngày 20/10/2023 - 23h59 ngày 29/10/2023 (hạn chót nộp bài)**

\*\*\* HV submit bài làm qua Google Classroom , mục bài thi cuối khóa hoặc gửi bài thi qua email [eric.le2131@gmail.com](mailto:eric.le2131@gmail.com) (khuyến khích cách 1) \*\*\*

\*\*\* HV sẽ bị trừ điểm nếu bài làm giống nhau \*\*\*

\*\*\* HV phải gửi bài làm đúng hạn nộp bài, sau hạn nộp bài nếu HV không gửi thì sẽ không được chấm điểm \*\*\*

**Chú ý, với mỗi câu :**

- Lần lượt thực hiện các bước làm bài theo quy trình đã được hướng dẫn làm demo/bài tập trong lớp.
- Mỗi câu được làm trong 1 file jupyter notebook riêng biệt. Trong trường hợp câu hỏi có yêu cầu giải thích/ trả lời bằng từ ngữ, học viên tạo 1 mark down cell để ghi lại câu trả lời.
- Với các câu hỏi yêu cầu đưa ra nhận xét, học viên nhận xét chi tiết nhất có thể.
- Câu nào có phần trực quan hóa kết quả thì vừa phải trực quan vừa phải giải thích.

**Object Detection : Detect Logo In Image (20 pts)**

Cho tập dữ liệu Logo Detection trong file logo\_dataset.zip

Yêu cầu :

1. Load tập data trên, chia tập data trên thành 2 tập con train/valid theo tỉ lệ 80/20
2. Tiến hành khởi tạo 1 mô hình YOLOv8, phiên bản YOLOv8s (small) hoặc YOLOv8 (medium)
3. Tiến hành huấn luyện model YOLOv8 đã tạo ở bên trên với tập data hiện có. Report performance của model trên tập valid. Nhận xét
4. Dựa vào link ở đây : [Hyperparameter Tuning - Ultralytics YOLOv8 Docs](#) , hãy thử nghiệm tối thiểu 1 cách tuning hyperparameter cho model YOLOv8. Report kết quả và nhận xét.

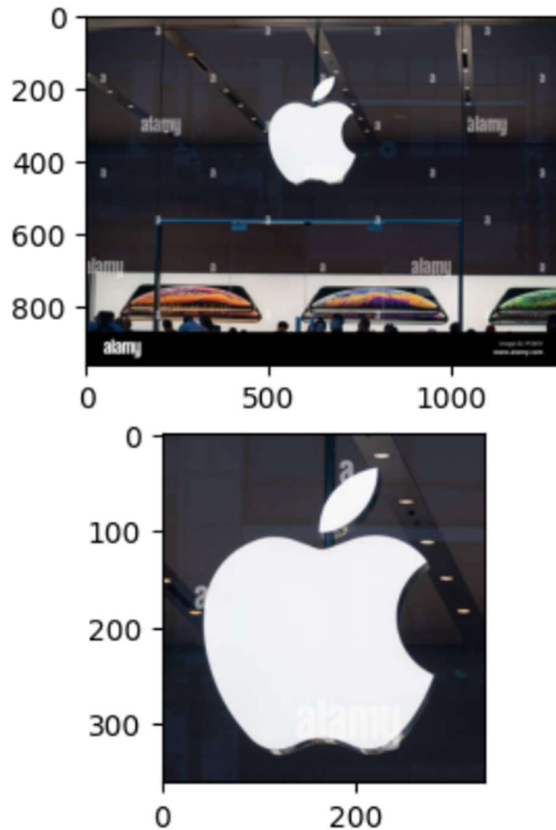
**Image Classification : Logo Classification (30 pts)**

Cho tập dữ liệu dùng để nhận dạng Logo, học viên tiến hành download bộ data trên bằng cách sử dụng code trong file `data_download.ipynb`. Sau khi download, học viên tiến hành đọc dataframe của bộ data từ file « /flickr\_logos\_27\_dataset/flickr\_logos\_27\_dataset\_training\_set\_annotation.txt »

Yêu cầu :

1. Load tập data trên, chia tập data trên thành 3 tập con train /test theo tỉ lệ 80/20
2. Tiến hành khởi tạo 1 mô hình CNN, học viên có thể tự tạo 1 mô hình CNN hoàn toàn mới hoặc sử dụng các Pretrained Model
3. Tiến hành huấn luyện model đã tạo ở bên trên với tập data hiện có. Report performance của model trên tập test. Nhận xét
4. Từ model YOLOv8 đã thu được ở câu Object Detection, tiến hành ghép nối với model Classification hiện tại, để tạo thành 1 pipeline : `Image => YOLOv8 => Cropped Logo Images => Classification Model => Logo Recognition`. Với câu hỏi này, học viên tạo 1 hàm để ghép nối 2 model thành 1 pipeline.

5. Sử dụng pipeline trên để dự đoán trên các hình ảnh trong tập tin *test\_cases.zip*. Visualize ảnh gốc, ảnh được model YOLO crop, và kết quả nhận diện logo, ví dụ như hình dưới.



```
tensor([1], device='cuda:0')  
[151]: array(['Apple'], dtype=object)
```

### Generative AI : GAN (15 pts)

Cho tập dữ liệu FASHION MNIST từ Pytorch:

<https://pytorch.org/vision/stable/generated/torchvision.datasets.FashionMNIST.html>

Yêu cầu :

1. Tận dụng kiến thức về các mô hình GAN đã học, học viên hãy phát triển 1 mô hình GAN có thể tạo ra các ảnh tương tự như ảnh trong tập dataset FASHION MNIST
2. Visualize và đánh giá kết quả thu được.

## Face Recognition (25 pts)

Cho tập dataset data\_2\_final.zip, chứa các hình ảnh gương mặt cần nhận diện của 100 người (identity) khác nhau.

File data.csv gồm 2 cột image – chứa tên ảnh, và label – chứa mã số identity của ảnh. Các ảnh có cùng label thuộc về cùng 1 người (identity).

Vận dụng những kiến thức đã học về Face Recognition, học viên hãy thực hiện các yêu cầu sau :

1. Sử dụng kiến trúc mô hình GhostFaceNet (2023) tiến hành xây dựng 1 mô hình nhận diện khuôn mặt cho tập dataset trên, sử dụng các pretrained weight sẵn có (học viên có thể tùy chọn pretrained weight). Đánh giá hiệu quả của mô hình trên tập data này thông qua Accuracy, True Positive Rate, False Positive Rate. Yêu cầu : Độ chính xác cao >98%.
2. Học viên tiến hành mô phỏng tình huống : Đăng ký thêm 1 identity vào ứng dụng Face Recognition sẵn có bằng việc thực hiện các bước sau :
  - a. Sử dụng kiến thức về Aggregation Enhanced Embedding Vector, tiến hành tạo 1 vector đại diện cho mỗi identity (ví dụ : 100 identity sẽ có 100 vector đại diện)
  - b. Giả sử khi user mới đăng ký trong hệ thống, cần submit 2 tấm ảnh chân dung. Học viên hãy tự chọn 2 ảnh chân dung thuộc cùng 1 người bất kỳ (bản thân, bạn bè...) và tiến hành khởi tạo vector đại diện cho user mới này, add các thông tin về ID, filenames của 2 ảnh chân dung vào bộ data sẵn có.
3. Học viên tiến hành mô phỏng tình huống : 1 User login vào hệ thống Face Recognition, bằng cách submit 1 tấm ảnh chân dung (lưu ý ảnh chân dung dùng để test trong câu hỏi này không được trùng với 2 ảnh đã submit ở câu hỏi 2, và phải thuộc cùng 1 identity). Tìm ID của ảnh vừa submit, kèm theo visualize các ảnh chân dung của ID đó trong dataset.

## Application Deployment (10 pts)

Vận dụng những kiến thức đã học, học viên hãy triển khai model Face Recognition ở câu Face Recognition trên nền tảng Cloud AWS, dưới dạng Gradio APP. Với câu hỏi này, học viên gửi link tới Web App của APP hoặc gửi link video clip demo APP trên localhost.

Ví dụ Demo : <http://13.239.2.162:7860/>

Yêu cầu:

1. Cho phép user upload 3 tấm ảnh gương mặt để đăng ký nhận dạng, kèm theo tên.
2. Mỗi khi user check in, yêu cầu user upload một tấm ảnh, và trả về tên của người vừa check in.
3. Ứng dụng phải được chạy trên nền tảng cloud, sử dụng Gradio.

**--- Chúc các bạn làm bài tốt ---**