**CS-300 Artificial Intelligence**

**Final Project**

***Dr. Nguyen Ngoc Thao***

***Msc. Do Trong Le – Nguyen Quang Thuc***

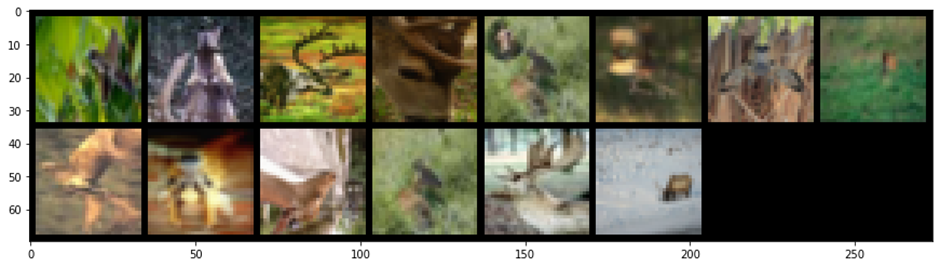
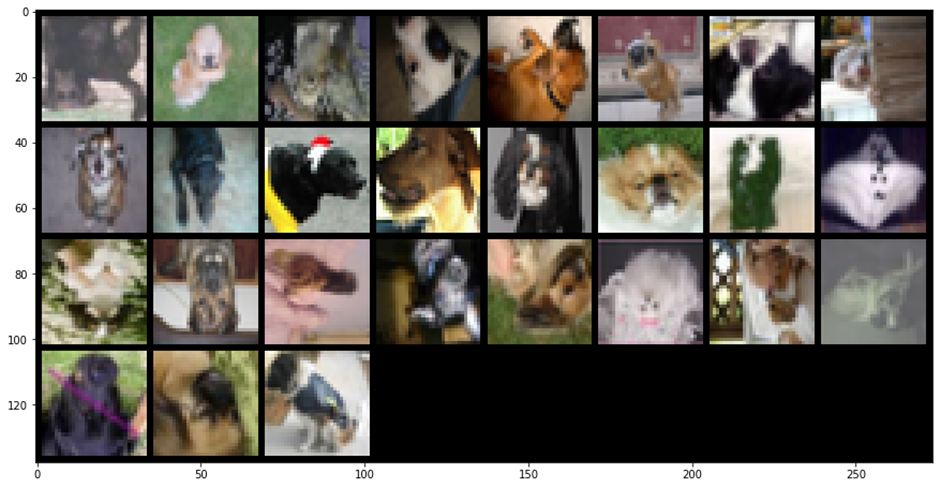
**Hình thức:** Nhóm tối đa 4 sinh viên

**Hạn nộp: 31/3**

## Đề bài:

Trong dự án này, sinh viên sẽ thực hiện xây dựng một hệ thống nhận diện hình ảnh đảo ngược bằng cách sử dụng các mô hình học máy. Cụ thể, đầu vào của mô hình AI sẽ là một tấm ảnh, mô hình phải xác định xem ảnh đó được lật ngược theo chiều dọc (vertical flip) hay không.

## Yêu cầu cần thực hiện:

* **1) Chọn dataset và tạo dữ liệu**: Sinh viên cần lựa chọn ***một dataset*** (vd: CIFAR-10, STL-10, MNIST, MNIST-Fashion) chứa các hình ảnh tự nhiên từ Hugging Face Hub hoặc các nguồn tương tự. Sau đó, các bạn sẽ tạo dữ liệu với nhãn mới một cách **tự động** (ví dụ ảnh được lật có nhãn là 1, ảnh không được lật có nhãn là 0). *Sinh viên có thể lật ảnh ngẫu nhiên trong quá trình training hoặc tạo trước dữ liệu nhưng* ***lưu ý về tính cân bằng*** *giữa số lượng ảnh được lật và ảnh không lật.* Dữ liệu sau khi sinh ra cần phải được chia thành 2 tập trainset - testset với tỉ lệ 8:2. Dưới dây là ảnh ví dụ về các ảnh con chó và con hươu được lật và không lật, ảnh gốc được lấy từ trong tập CIFAR-10
* 
* **2) Xây dựng mô hình:** Sinh viên sử dụng một framework học máy như Scikit-learn, PyTorch, hoặc các framework tương tự để xây dựng ***hai mô hình*** học máy khác họ (ví dụ Resnet-50 và Resnet-101 là cùng họ) để phân loại hình ảnh theo hướng đúng hoặc đảo ngược. *Lưu ý, sinh viên có thể dùng các mô hình đơn giản như DecisionTree, SVM được cung cấp bởi Scikit-learn để đơn giản trong việc tùy biến các tham số.*
* **3) Huấn luyện mô hình:** Sinh viên cần huấn luyện mô hình trên trainset và điều chỉnh các tham số mô hình để đạt được độ chính xác mong muốn *(ít nhất có 1 mô hình đạt kết quả trên 80%)*
* **4) Đánh giá:** Sau khi huấn luyện xong, sinh viên cần đánh giá hiệu suất của mô hình trên testset và tinh chỉnh mô hình nếu cần. Việc đánh giá hiệu suất bao gồm kiểm tra độ chính xác mô hình, chỉ ra những trường hợp mô hình không hoạt động tốt (thường thuộc vào lớp nào, hoặc những hình đó có gì đặc biệt).

## 

## Mục chấm điểm:

* **Code (50%)**: Chất lượng code và cấu trúc của mã nguồn.
  + Cài đặt việc tạo dữ liệu upside-down (40%)
  + Cài đặt mô hình và huấn luyện mô hình (40%)
  + Đánh giá mô hình dựa trên accuracy (10%)
  + In ra 10 hình dự đoán sai (10%) của 3 lớp có số dự đoán sai nhiều nhất, của dữ liệu ban đầu (ví dụ: dữ liệu CIFAR-10 có 10 lớp ban đầu, chọn ra 3 lớp có accuracy thấp nhất và in ra hình dự đoán sai của 3 lớp trên)
* **Báo cáo (50%)**: Báo cáo về quá trình thực hiện dự án, bao gồm:
  + lựa chọn, và xây dựng dataset (30%)
  + kiến trúc mô hình (30%)
  + các thử nghiệm, kết quả đạt được (20%)
  + phân tích các trường hợp ảnh bị phân loại sai và đưa ra lý do (20%)

## Tài liệu tham khảo:

1. Cách **tạo một bộ dữ liệu** cho bài toán sử dụng pytorch: [[LINK]](https://pytorch.org/tutorials/beginner/basics/data_tutorial.html#creating-a-custom-dataset-for-your-files)
2. Cách **huấn luyện một mô hình** phân loại sử dụng pytorch: [[LINK]](https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html)
3. Sau khi làm quen với việc huấn luyện mô hình bằng mô hình đơn giản ở trên, sinh viên có thể tham khảo thêm cách **huấn luyện mô hình từ một mô hình đã được huấn luyện** bằng kĩ thuật transfer learning hoặc fine-tuning [[LINK]](https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html#finetuning-the-convnet)

*Lưu ý:*

* *sinh viên cần phải thêm hàm lật ảnh và thay đổi* ***label*** *cho phù hợp*
* *số lượng lớp cần phân loại trong project được yêu cầu là 2 (0 cho ảnh không lật, và 1 cho ảnh đã lật)*
* *sinh viên nhớ dùng lệnh* **model.eval()** *trước khi chạy với tập test. Sinh viên nên xem kĩ các link trong* ***tài liệu tham khảo*** *để rõ hơn cách in ảnh trên tập test. Cần chú ý, các hàm trong hướng dẫn* ***chưa bao gồm các yêu cầu trong project,*** *sinh viên cần thực hiện thêm các yêu cầu trong bài.*

*Gợi ý:*

* *sinh viên có thể trả về 2 label trong hàm* \_\_getitem\_\_ *để dễ in ra hình dự đoán sai của từng lớp trong yêu cầu báo cáo. Ví dụ:*

**def** \_\_getitem\_\_**(**self**,** **idx):**

***# TODO: get and transform both image and label with flip rate is 50%***

**return** **image,** [**flip\_label**](https://pytorch.org/docs/stable/tensors.html#torch.Tensor)**, label**

Lúc lấy dữ liệu, sinh viên chỉ cần lấy flip\_labels để huấn luyện mô hình:

**for** **i,** **data** **in** enumerate**(**[**trainloader**](https://pytorch.org/docs/stable/data.html#torch.utils.data.DataLoader)**,** 0**):**

*# get the inputs; data is a list of [inputs, flip\_labels, labels]*

**i**[**nputs**](https://pytorch.org/docs/stable/tensors.html#torch.Tensor)**,** [**flip\_labels**](https://pytorch.org/docs/stable/tensors.html#torch.Tensor), **labels** **=** **data**

Note: Sinh viên nên sử dụng pytorch version <2.0 và >1.7