

# Bài tập Deep Learning

Author Thanh-Ha DO

## 1 Bài tập 1: (Tập dữ liệu Imagenette2)

Bộ dữ liệu Imagenette2 được phát hành bởi nhóm fastai là một phiên bản nhỏ của ImageNet, được tạo ra để thử nghiệm các mô hình học sâu về phân loại ảnh mà không cần tải toàn bộ ImageNet. Đây là bộ dữ liệu hoàn toàn public *phục vụ cho học tập và giảng dạy* ([link](#)).

Trong bài tập này, học viên sử dụng phiên bản imagenette2 (ảnh kích thước 320x320 pixel).

Bộ dữ liệu gồm 10 lớp với nhãn là mã định danh dạng **n0xxxxxx**.

**Ví dụ:** Khi giải nén bộ dữ liệu, học viên có thể thấy một thư mục có tên n03394916. Đây là mã WNID tương ứng với lớp French Horn.

**Bảng ánh xạ giữa WNID và tên lớp:**

WNID	Tên lớp
n01440764	Tench
n02102040	English Springer
n02979186	Cassette Player
n03000684	Chain Saw
n03028079	Church
n03394916	French Horn
n03417042	Garbage Truck
n03425413	Gas Pump
n03445777	Golf Ball
n03888257	Parachute

Dành ra vài phút để đọc hiểu dữ liệu và thực hiện các yêu cầu sau trên tập train:

- Vẽ biểu đồ cột thống kê số lượng mẫu của mỗi lớp, đưa ra nhận xét
- Với mỗi lớp hiện có, hiển thị 3 ảnh và nhãn lên màn hình để trực quan hóa dữ liệu.
- Thực hiện tăng cường dữ liệu để làm đa dạng hóa bộ dữ liệu, hiển thị kết quả thu được. Học viên áp dụng ít nhất 3 kỹ thuật tăng cường dữ liệu dưới đây:
  - Lật ngang ảnh (Horizontal Flip).
  - Xoay ảnh ngẫu nhiên (Random Rotation từ -15 độ đến 15 độ).
  - Thay đổi độ sáng hoặc độ tương phản ảnh (Brightness/Contrast).

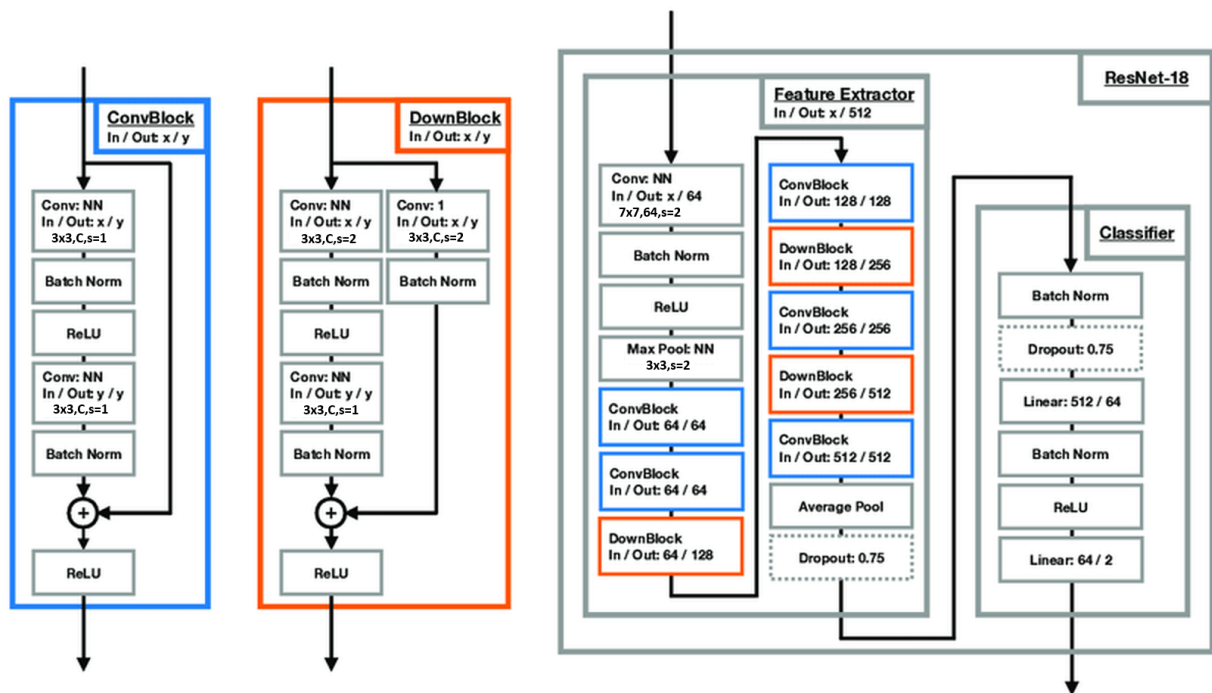
**Chú ý:** Sử dụng PyTorch hoặc TensorFlow để xây dựng Dataset/Dataloader, trình bày mã nguồn rõ ràng, có bình luận từng bước thực hiện.

## 2 Bài tập 2: (Kiến trúc ResNet)

ResNet (Residual Network) là một kiến trúc mạng nơ-ron sâu do Microsoft giới thiệu năm 2015, nổi bật với cơ chế kết nối tắt (skip connection). Cơ chế này giúp truyền trực tiếp đầu vào của một lớp đến đầu ra của lớp sau, giúp mô hình giảm hiện tượng mất thông tin và gradient vanishing khi mạng trở nên rất sâu.

Cho hình vẽ dưới đây và sử dụng framework bất kỳ, thực hiện yêu cầu sau:

- Dựa vào hình vẽ, xây dựng chính xác kiến trúc mạng ResNet-18. Học viên cần chú ý tham số các lớp ConvBlock, DownBlock, số lượng các lớp Convolution, kích thước kernel, stride và các skip connection giữa các tầng.
- Hiển thị kiến trúc của mô hình vừa xây dựng bằng các công cụ visualize phù hợp như:



- Với **PyTorch**, học viên có thể dùng thư viện như `torchsummary`, `torchviz` để hiển thị kiến trúc và flow của mạng.
- Với **TensorFlow**, học viên có thể dùng phương pháp `model.summary()` hoặc thư viện `TensorBoard` để hiển thị kiến trúc mạng.

(c) Chạy thử mô hình với dữ liệu đầu vào giả (dummy data) có kích thước đầu vào (`batchsize, 3, 224, 224`) để xác nhận mô hình hoạt động đúng, và hiển thị kích thước đầu ra từng tầng.

### 3 Bài tập 3: (Huấn luyện mô hình)

Sử dụng kiến trúc mạng ResNet-18 vừa xây dựng để thực hiện huấn luyện trên bộ dữ liệu Imagenette2-320.

- Tải dữ liệu vào Dataset/Dataloader (Thực hiện resize ảnh về 224x224 và chuẩn hóa).
- Lựa chọn loss function là `crossentropy`, hàm tối ưu Adam (learning rate =  $1e-3$ ) và batch size = 32.
- Huấn luyện mô hình trong tối thiểu 10 epochs. Theo dõi và lưu lại kết quả độ chính xác (accuracy) và mất mát (loss) trên tập huấn luyện và kiểm thử qua từng epoch.
- Vẽ đồ thị biểu diễn sự thay đổi của accuracy và loss qua các epochs. Đưa ra nhận xét về hiệu quả của việc huấn luyện (hiện tượng hội tụ, overfitting, underfitting,... nếu có) và cho biết các siêu tham số đã mang lại kết quả tối ưu nhất chưa.

**Yêu cầu:** Trình bày mã nguồn rõ ràng, chi tiết và có chú thích từng bước. Học viên có thể sử dụng GPU để hỗ trợ và tăng tốc huấn luyện.