

Báo cáo giải pháp: Phân loại ảnh Chó – Mèo bằng VGG

Trung Truong

August 23, 2025

1 Mục tiêu

Xây dựng hệ thống phân loại ảnh chó và mèo với độ chính xác cao, sử dụng các kiến trúc VGG (VGG11, VGG13, VGG16, VGG19) trên PyTorch, đồng thời triển khai được backend để nhận ảnh và dự đoán nhãn.

2 Quy trình giải pháp

2.1 Chuẩn bị dữ liệu

- Dataset: `cats_and_dogs_filtered.zip` gồm tập train và validation.
- Chia tập train thành train/validation theo tỷ lệ 90%/10%.
- Resize ảnh về (224, 224), normalize với mean/std chuẩn của ImageNet.

2.2 Augmentation (chỉ tập train)

- Random rotation $\pm 5^\circ$
- Random horizontal flip 50%
- Random crop với padding 10

Mục tiêu: giảm overfitting, tăng khả năng tổng quát của model.

2.3 Kiến trúc mô hình

- Sử dụng VGG với batch normalization.
- Classifier gồm 3 lớp fully-connected với dropout 0.5.
- Output dimension: 2 (Dog/Mèo).

2.4 Huấn luyện

- Loss function: CrossEntropyLoss
- Optimizer: Adam, learning rate $5e-4$ (classifier) và $5e-5$ (feature layers)
- Batch size: 32
- Epochs: 5
- Lưu model tốt nhất dựa trên validation loss

2.5 Đánh giá

- Tính accuracy trên tập test.
- Theo dõi các ví dụ dự đoán sai để phân tích.

3 Kết quả

Model	Test Loss	Test Accuracy
VGG11	0.604	68.36%
VGG13	0.666	58.59%
VGG16	0.729	51.56%
VGG19	0.758	49.71%

Table 1: Kết quả đánh giá các model trên tập test

Nhận xét: VGG11 đạt accuracy cao nhất; các model sâu hơn có thể overfitting do dữ liệu hạn chế.

4 Vấn đề tồn tại

- Dataset nhỏ, dễ gây overfitting với các model VGG sâu (VGG16, VGG19).
- Accuracy chưa cao (khoảng 68% cho model tốt nhất).
- Chưa thử các kỹ thuật transfer learning từ ImageNet.

5 Ý tưởng cải tiến

- Sử dụng pretrained VGG trên ImageNet để fine-tune, giảm overfitting.
- Tăng cường dữ liệu với nhiều augmentation hơn (color jitter, cutout, etc.).
- Thử các kiến trúc nhẹ hơn hoặc các CNN hiện đại (ResNet, EfficientNet).
- Sử dụng kỹ thuật early stopping, learning rate scheduler để cải thiện hiệu năng.