HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CƠ BẢN I

THỰC TẬP CƠ SỞ

BÁO CÁO GIỮA KỲ

Đề tài:

Xây dựng và huấn luyện mô hình YOLOv3

Giảng viên hướng dẫn	**:Kim Ngọc Bách**
Họ và tên sinh viên	**:Dương Đăng Khoa**
Mã sinh viên	**:B23DCCN441**
Lớp	**:D23CQCN07-B**
Số điện thoại	**:0354469232**
Ngày sinh	**:06/03/2005**

Hà Nội – 2026

1. GIỚI THIỆU DỰ ÁN

1.1. Lý do chọn đề tài

Trong những năm gần đây, bài toán phát hiện đối tượng (Object Detection) đã trở thành một trong những hướng nghiên cứu trọng tâm của lĩnh vực Thị giác máy tính (Computer Vision). Sự phát triển mạnh mẽ của Deep Learning, đặc biệt là các mạng Convolutional Neural Network (CNN), đã giúp các mô hình phát hiện đối tượng đạt độ chính xác và tốc độ xử lý cao.

Trong số các phương pháp phát hiện đối tượng hiện nay, các mô hình thuộc nhóm one-stage detector như YOLO (You Only Look Once) được đánh giá cao nhờ khả năng xử lý nhanh và hiệu quả trong thời gian thực. Đặc biệt, YOLOv3 là phiên bản cải tiến quan trọng, có sự cân bằng tốt giữa tốc độ và độ chính xác, đồng thời được ứng dụng rộng rãi trong thực tế.

Việc lựa chọn đề tài "Xây dựng và huấn luyện mô hình YOLOv3 cho bài toán phát hiện đối tượng" xuất phát từ các lý do sau:

- YOLOv3 là mô hình tiêu biểu, có kiến trúc rõ ràng và mang tính đại diện cao trong nhóm one-stage detector.
- Mô hình kết hợp giữa backbone Darknet-53 và cơ chế dự đoán đa tỉ lệ (multi-scale prediction), giúp nâng cao khả năng phát hiện đối tượng ở nhiều kích thước khác nhau.
- Đề tài giúp sinh viên hiểu sâu về cấu trúc mạng CNN, residual learning, cơ chế anchor box, loss function và các chỉ số đánh giá như IoU, Precision, Recall và mAP.
- Việc tự xây dựng và huấn luyện mô hình từ đầu giúp nâng cao kỹ năng lập trình, xử lý dữ liệu, tối ưu hóa mô hình và đánh giá thực nghiệm.

Thông qua đề tài này, người thực hiện không chỉ củng cố kiến thức lý thuyết về Deep Learning và thị giác máy tính mà còn rèn luyện kỹ năng triển khai mô hình, phân tích kết quả và đánh giá hiệu năng hệ thống.

1.2.Ý nghĩa và tính ứng dụng

1.2.1. Ý nghĩa khoa học

Đề tài "Xây dựng và huấn luyện mô hình YOLOv3 cho bài toán phát hiện đối tượng" mang ý nghĩa khoa học quan trọng trong lĩnh vực Thị giác máy tính và Học sâu (Deep Learning).

Thứ nhất, đề tài giúp làm rõ nguyên lý hoạt động của các mô hình phát hiện đối tượng thuộc nhóm one-stage detector, đặc biệt là cơ chế dự đoán đa tỉ lệ (multi-scale prediction), anchor box và hàm mất mát đặc trưng của YOLOv3. Việc nghiên cứu và triển khai mô hình từ đầu giúp hiểu sâu cấu trúc mạng CNN, residual learning và quá trình lan truyền ngược (backpropagation).

Thứ hai, đề tài góp phần củng cố kiến thức về các chỉ số đánh giá mô hình phát hiện đối tượng như IoU, Precision, Recall và mAP. Việc thực nghiệm và phân tích kết quả giúp người thực hiện có cái nhìn toàn diện về hiệu năng mô hình trong các điều kiện dữ liệu khác nhau.

Thứ ba, đề tài tạo nền tảng cho việc mở rộng nghiên cứu sang các phiên bản YOLO mới hơn hoặc các mô hình khác như RetinaNet, Faster R-CNN, SSD,… từ đó so sánh và đánh giá hiệu quả giữa các phương pháp.

1.2.2. Ý nghĩa thực tiễn

Phát hiện đối tượng là một trong những bài toán cốt lõi của thị giác máy tính và có tính ứng dụng cao trong nhiều lĩnh vực của đời sống và sản xuất. Việc nghiên cứu và triển khai thành công mô hình YOLOv3 có thể được ứng dụng trong các hệ thống thực tế yêu cầu xử lý nhanh và chính xác.

Đề tài giúp người thực hiện nắm được quy trình xây dựng một hệ thống AI hoàn chỉnh, bao gồm tiền xử lý dữ liệu, huấn luyện mô hình, đánh giá kết quả và trực quan hóa dự đoán. Đây là những kỹ năng quan trọng trong môi trường công nghiệp hiện nay.

1.2.3. Ứng dụng của mô hình YOLOv3

Mô hình YOLOv3 có thể ứng dụng rộng rãi trong nhiều lĩnh vực như sau:

- **Giao thông thông minh**
  - Phát hiện và phân loại phương tiện (ô tô, xe máy, xe tải, xe buýt…)
  - Giám sát và phát hiện vi phạm giao thông (vượt đèn đỏ, sai làn đường…)
  - Ứng dụng trong hệ thống hỗ trợ lái xe nâng cao (ADAS)
- **Công nghiệp**
  - Kiểm tra lỗi sản phẩm tự động trên dây chuyền sản xuất
  - Phát hiện vật thể nguy hiểm trong môi trường làm việc
  - Ứng dụng trong robot công nghiệp để nhận diện và phân loại vật thể
- **An ninh – giám sát**
  - Nhận diện và theo dõi con người trong khu vực giám sát
  - Phát hiện hành vi bất thường
  - Xây dựng hệ thống camera giám sát thông minh
- **Y tế**
  - Phát hiện tổn thương hoặc bất thường trong ảnh y khoa
  - Hỗ trợ phân tích ảnh X-ray, CT, MRI

- **Thương mại**
  - Nhận diện sản phẩm trong cửa hàng bán lẻ
  - Ứng dụng trong hệ thống thanh toán tự động

1.3. Giá trị học thuật

Đề tài này có giá trị học thuật cao vì giúp nghiên cứu các khía cạnh sau:

➢ **Kiến trúc mạng sâu**

- Phân tích backbone Darknet-53
- Residual learning
- Feature Pyramid Network (FPN)
- Multi-scale detection

➢ **Thuật toán tối ưu**

- Loss function cho bounding box regression
- Objectness loss
- Class prediction loss
- Anchor matching strategy

➢ **Các chỉ số đánh giá**

- Intersection over Union (IoU)
- Precision – Recall
- mean Average Precision (mAP)

1. CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG

2.1. Cơ sở lý thuyết

2.1.1 Tổng quan về Object Detection

Phát hiện đối tượng (Object Detection) là bài toán trong lĩnh vực Thị giác máy tính nhằm:

- Xác định vị trí của đối tượng trong ảnh (bounding box)
- Phân loại đối tượng thuộc lớp nào

Bài toán này kết hợp hai nhiệm vụ:

- Localization (xác định vị trí)
- Classification (phân loại)

Hiện nay, các phương pháp phát hiện đối tượng được chia thành hai nhóm chính:

- **Two-stage detectors**
  - Ví dụ: Faster R-CNN
  - Hoạt động qua hai bước: đề xuất vùng (Region Proposal) và phân loại

    - Độ chính xác cao nhưng tốc độ chậm
- **One-stage detectors**
    - Ví dụ: YOLOv3, SSD
    - Dự đoán trực tiếp bounding box và class trong một lần forward
    - Tốc độ cao, phù hợp thời gian thực

2.1.2. Mạng nơ-ron tích chập (CNN)

YOLOv3 được xây dựng dựa trên **Convolutional Neural Network (CNN)**.

CNN gồm các thành phần chính:
- Convolution layer
- Activation function (ReLU, LeakyReLU)
- Pooling layer
- Fully Connected layer

CNN có khả năng:
- Trích xuất đặc trưng không gian
- Nhận diện mẫu trong ảnh
- Học biểu diễn phân cấp từ thấp đến cao

2.1.3. Kiến trúc YOLOv3

Mô hình YOLOv3 bao gồm:

a) Backbone – Darknet-53
YOLOv3 sử dụng backbone Darknet-53 để trích xuất đặc trưng.
Darknet-53:
- Gồm 53 lớp tích chập
- Sử dụng Residual Block
- Giúp giảm hiện tượng vanishing gradient
- Tăng khả năng học sâu

b) Multi-scale Detection
YOLOv3 dự đoán ở 3 scale:
- 13×13 (phát hiện vật thể lớn)
- 26×26 (vật thể trung bình)
- 52×52 (vật thể nhỏ)

Cơ chế này giúp mô hình:
- Phát hiện tốt vật thể ở nhiều kích thước khác nhau
- Cải thiện độ chính xác

c) Anchor Boxes
Anchor box là các khung cố định được sử dụng để:
- Hỗ trợ dự đoán bounding box
- Xử lý vật thể có tỉ lệ khác nhau

Mỗi grid cell sẽ dự đoán nhiều anchor.

2.1.4. Các chỉ số đánh giá

a) Intersection over Union (IoU)
IoU đo mức độ chồng lấp giữa:
- Bounding box dự đoán
- Bounding box ground truth

$$IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union}$$

b) Precision và Recall
- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)

c) mean Average Precision (mAP)
mAP là trung bình của Average Precision trên tất cả các lớp.

Đây là thước đo tiêu chuẩn để đánh giá mô hình detection.

2.2. Công nghệ sử dụng

2.2.1. Ngôn ngữ lập trình
- **Python**

Python được lựa chọn là ngôn ngữ lập trình chính trong đề tài nhờ những ưu điểm nổi bật trong lĩnh vực Trí tuệ nhân tạo và Học sâu (Deep Learning). Python có cú pháp đơn giản, dễ đọc và dễ triển khai, giúp rút ngắn thời gian phát triển và thử nghiệm mô hình.

Bên cạnh đó, Python sở hữu hệ sinh thái thư viện phong phú hỗ trợ xử lý dữ liệu, tính toán khoa học, thị giác máy tính và học máy. Đây là ngôn ngữ được sử dụng phổ biến trong cộng đồng nghiên cứu AI toàn cầu, với tài liệu hướng dẫn đầy đủ và cộng đồng hỗ trợ lớn.

2.2.2. Thư viện Deep Learning
- **Pytorch**

PyTorch là thư viện Deep Learning được sử dụng để xây dựng và huấn luyện mô hình YOLOv3 trong đề tài. PyTorch cung cấp cơ chế tính toán tự động đạo hàm (Autograd), giúp việc xây dựng và tối ưu hàm mất mát trở nên thuận tiện và linh hoạt.

Thư viện này hỗ trợ xây dựng kiến trúc mạng nơ-ron thông qua các module như `nn.Module`, `nn.Conv2d`, `nn.BatchNorm2d`, `nn.Upsample`,… phù hợp với cấu trúc của Darknet-53 và các Detection Block trong YOLOv3.

Ngoài ra, PyTorch hỗ trợ tăng tốc tính toán trên GPU thông qua CUDA, giúp rút ngắn đáng kể thời gian huấn luyện mô hình so với việc chạy trên CPU.

PyTorch cũng cho phép dễ dàng kiểm tra, chỉnh sửa và debug mô hình trong quá trình phát triển nhờ cơ chế dynamic computation graph

2.2.3. Thư viện xử lí ảnh

- **OpenCV**
  OpenCV là thư viện xử lý ảnh và thị giác máy tính phổ biến. Trong đề tài, OpenCV được sử dụng để:

- Đọc và ghi ảnh
- Resize ảnh về kích thước phù hợp với mô hình
- Vẽ bounding box và nhãn lớp lên ảnh
- Hỗ trợ kiểm tra kết quả dự đoán (visualization)

- **Pillow (PIL)**
  Pillow là thư viện hỗ trợ xử lý ảnh trong Python, thường được sử dụng cùng với PyTorch. Thư viện này hỗ trợ:
- Mở và chuyển đổi định dạng ảnh
- Chuyển ảnh sang tensor
- Thực hiện một số thao tác tiền xử lý đơn giản

- **Matplotlib**
  Matplotlib được sử dụng để:
- Vẽ biểu đồ loss trong quá trình huấn luyện
- Vẽ biểu đồ Precision–Recall
- Minh họa kết quả thực nghiệm

2.2.4. Dataset sử dụng

- **Pascal VOC**

  Pascal VOC (Visual Object Classes) là bộ dữ liệu phổ biến trong bài toán phát hiện đối tượng. Bộ dữ liệu này bao gồm 20 lớp đối tượng như: person, car, dog, cat, chair, bottle,… cùng với nhãn bounding box cho từng đối tượng trong ảnh.

  Trong phạm vi đề tài, bộ dữ liệu Pascal VOC được lựa chọn vì:
- Số lượng lớp vừa phải (20 classes), phù hợp cho môi trường học thuật
- Kích thước dữ liệu không quá lớn, dễ huấn luyện
- Được sử dụng rộng rãi làm chuẩn đánh giá trong nhiều nghiên cứu
- Phù hợp để đánh giá mAP@0.5 theo chuẩn VOC

2. PHÂN TÍCH YÊU CẦU DỰ ÁN

3.1 . Mục tiêu của dự án

Dự án nhằm xây dựng và huấn luyện mô hình YOLOv3 để thực hiện bài toán phát hiện đối tượng trong ảnh

Cụ thể

- Xây dựng kiến trúc YOLOv3 từ đầu bằng PyTorch
- Huấn luyện mô hình trên bộ dữ liệu (ví dụ: Pascal VOC)
- Đánh giá hiệu năng bằng các chỉ số IoU, Precision, Recall, mAP
- Phân tích kết quả và đề xuất hướng cải tiến

3.2 . Yêu cầu chức năng

Hệ thống cần đáp ứng các yêu cầu sau:

- Đọc và xử lý dữ liệu ảnh và nhãn (annotation)
- Xây dựng mô hình YOLOv3 gồm:
- Huấn luyện mô hình với tập dữ liệu huấn luyện
- Lưu và tải trọng số mô hình (model checkpoint)
- Dự đoán (inference) trên ảnh mới
- Hiển thị bounding box và nhãn lớp
- Tính toán và hiển thị mAP

3.4 Yêu cầu về dữ liệu

- Dataset phải có:
    - Ảnh đầu vào
    - Annotation bounding box
    - Nhãn lớp tương ứng
- Dữ liệu được chia thành:
    - Tập huấn luyện (Training set)
    - Tập kiểm tra (Validation/Test set)

3. Kế hoạch thực hiện dự án

Giai đoạn 1 (Tuần 1–2): Nghiên cứu lý thuyết và chuẩn bị dữ liệu
Nội dung thực hiện:

- Nghiên cứu tổng quan về Object Detection
- Tìm hiểu kiến trúc YOLOv3 và backbone Darknet-53
- Tìm hiểu các chỉ số đánh giá: IoU, Precision, Recall, mAP
- Lựa chọn dataset (ví dụ: Pascal VOC)
- Chuyển đổi annotation về format YOLO
- Xây dựng pipeline đọc dữ liệu (Dataset, DataLoader)

Giai đoạn 2 (Tuần 3–4): Xây dựng kiến trúc mô hình

Nội dung thực hiện:

- Cài đặt ConvBlock và ResidualBlock
- Xây dựng Darknet-53
- Thiết kế Detection Block
- Kết hợp backbone và head theo kiến trúc YOLOv3
- Kiểm tra forward pass và kích thước output

Giai đoạn 3 (Tuần 5–6): Xây dựng loss function và huấn luyện mô hình
Nội dung thực hiện:

- Cài đặt IoU function
- Xây dựng YOLO loss:
  - Bounding box regression loss
  - Objectness loss
  - Classification loss
- Thiết kế target tensor
- Huấn luyện mô hình trên training set
- Lưu checkpoint và theo dõi loss

Giai đoạn 4 (Tuần 7–8): Đánh giá và phân tích kết quả
Nội dung thực hiện:

- Cài đặt tính mAP
- Tính Precision – Recall
- Thực hiện inference trên tập test
- Phân tích kết quả đạt được
- So sánh với mô hình pretrained (nếu có)
- Đánh giá ưu điểm và hạn chế

Giai đoạn 5 (Tuần 9–10): Hoàn thiện và chỉnh sửa báo cáo cuối kỳ
Nội dung thực hiện:

- Tổng hợp toàn bộ nội dung đã thực hiện
- Chỉnh sửa bố cục và định dạng báo cáo
- Bổ sung bảng biểu, hình ảnh minh họa
- Viết phần kết luận và hướng phát triển
- Kiểm tra lại code và kết quả thực nghiệm

Tài liệu tham khảo

Everingham, M., Van Gool, L., Williams, C., Winn, J., & Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) challenge. *International Journal of Computer Vision*, *88*, 303–338. https://doi.org/10.1007/s11263-009-0275-4

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition* (arXiv:1512.03385). arXiv. https://doi.org/10.48550/arXiv.1512.03385

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2018). *Focal Loss for Dense Object Detection* (arXiv:1708.02002). arXiv. https://doi.org/10.48550/arXiv.1708.02002

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). *SSD: Single Shot MultiBox Detector* (Vol. 9905, pp. 21–37). https://doi.org/10.1007/978-3-319-46448-0_2

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, Real-Time Object Detection* (arXiv:1506.02640). arXiv. https://doi.org/10.48550/arXiv.1506.02640

Redmon, J., & Farhadi, A. (2016). *YOLO9000: Better, Faster, Stronger* (arXiv:1612.08242). arXiv. https://doi.org/10.48550/arXiv.1612.08242

Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement* (arXiv:1804.02767). arXiv. https://doi.org/10.48550/arXiv.1804.02767

Ren, S., He, K., Girshick, R., & Sun, J. (2016). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks* (arXiv:1506.01497). arXiv. https://doi.org/10.48550/arXiv.1506.01497