A

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÀI TẬP LỚN**

**HỌC PHẦN: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**

**Đề tài số 4: Xây dựng hệ thống nhận diện giới tính và tuổi sử dụng OpenCV**

**Giảng viên hướng dẫn: Lương Thị Hồng Lan**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | 20210612 | Nguyễn Văn Mạnh | DCCNTT12.10.2 |
| **2** | 20210620 | Dương Thị Mai Phương | DCCNTT12.10.2 |
| **3** | 20210435 | Phạm Quốc Hoàng | DCCNTT12.10.2 |
| **4** | 20210441 | Nguyễn Thanh Tùng | DCCNTT12.10.2 |

|  |  |
| --- | --- |
|  |  |

**Bắc Ninh, năm 2024**

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á  **KHOA CÔNG NGHỆ THÔNG TIN** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ 1, NĂM HỌC 2024** – **2025** |

|  |  |
| --- | --- |
| **PHIẾU CHẤM THI BÀI TẬP LỚN KẾT THÚC HỌC PHẦN**  **Mã đề thi: 4**  **Tên học phần: Xử lý ảnh và thị giác máy tính**  **Lớp Tín chỉ: XATGMT.03.K12.02.LH.C04.1\_LT** | |
| **Cán bộ chấm thi 1**  *(Ký và ghi rõ họ tên)* | **Cán bộ chấm thi 2**  *(Ký và ghi rõ họ tên)* |

| **TT** | **TIÊU CHÍ** | **THANG ĐIỂM** | Nguyễn Văn Mạnh | Dương Thị Mai Phương | Phạm Quốc Hoàng | Nguyễn Thanh Tùng |
| --- | --- | --- | --- | --- | --- | --- |
| 20210612 | 20210620 | 20210435 | 20210441 |
| **1** | **Nội dung báo cáo trên Word đầy đủ** | **3.5** |  |  |  |  |
| 1.1 | Có bố cục rõ ràng (mục lục, phần mở đầu, nội dung chính, kết luận). | 0,5 |  |  |  |  |
| 1.2 | Nội dung phân tích rõ ràng, logic. | 0,5 |  |  |  |  |
| 1.3 | Có dẫn chứng, số liệu minh họa đầy đủ. | 0,5 |  |  |  |  |
| 1.4 | Ngôn ngữ và trình bày chuẩn, không lỗi chính tả. | 0,5 |  |  |  |  |
| 1.5 | Có trích dẫn tài liệu tham khảo đúng quy cách. | 0,5 |  |  |  |  |
| 1.6 | Được trình bày chuyên nghiệp (canh lề, font chữ, khoảng cách dòng hợp lý). | 0,5 |  |  |  |  |
| 1.7 | Tài liệu đầy đủ, bám sát yêu cầu của đề bài. | 0,5 |  |  |  |  |
| **2** | **Nội dung thuyết trình đầy đủ** | **1.0** |  |  |  |  |
| 2.1 | Trình bày tự tin, phát âm rõ ràng, mạch lạc. | 0,5 |  |  |  |  |
| 2.2 | Nội dung thuyết trình đúng trọng tâm, không lan man. | 0,5 |  |  |  |  |
| **3** | **Slides báo cáo đầy đủ nội dung + Hỏi đáp** | **3.0** |  |  |  |  |
| 3.1 | Slides có bố cục rõ ràng (mở đầu, nội dung, kết luận). | 0,5 |  |  |  |  |
| 3.2 | Thiết kế slides đẹp, chuyên nghiệp (màu sắc, hình ảnh minh họa). | 0,5 |  |  |  |  |
| 3.3 | Nội dung trên slides ngắn gọn, dễ hiểu, súc tích. | 0,5 |  |  |  |  |
| 3.4 | Nội dung slides phù hợp với nội dung báo cáo. | 0,5 |  |  |  |  |
| 3.5 | Trả lời câu hỏi đầy đủ, chính xác. | 0,5 |  |  |  |  |
| 3.6 | Trả lời câu hỏi tự tin, thuyết phục. | 0,5 |  |  |  |  |
| **4** | **Code đầy đủ** | **2.5** |  |  |  |  |
| 1.1 | Code được trình bày rõ ràng, có chú thích đầy đủ. | 0,5 |  |  |  |  |
| 1.2 | Code chạy đúng, không lỗi. | 0,5 |  |  |  |  |
| 1.3 | Code tối ưu, không dư thừa. | 0,5 |  |  |  |  |
| 1.4 | Đáp ứng đầy đủ các yêu cầu chức năng theo đề bài. | 0,5 |  |  |  |  |
| 1.5 | Có tính sáng tạo hoặc cải thiện so với yêu cầu. | 0,5 |  |  |  |  |
| **TỔNG ĐIỂM BẰNG SỐ:** | | **10** |  |  |  |  |
| **TỔNG ĐIỂM BẰNG CHỮ:** | | *Mười tròn* |  |  |  |  |

**MỤC LỤC**

[LỜI MỞ ĐẦU 7](#_Toc184607949)

[CHƯƠNG I: TỔNG QUAN 9](#_Toc184607950)

[1.1. Nhận dạng đối tượng 9](#_Toc184607951)

[1.1.1. Định nghĩa nhận dạng đối tượng 9](#_Toc184607952)

[1.1.2. Những vấn đề cần giải quyết trong bài tập nhận dạng 10](#_Toc184607953)

[1.1.3. Ứng dụng của bài toán nhận dạng đối tượng 13](#_Toc184607954)

[1.2 Các phương pháp áp dụng cho bài toán nhận dạng 14](#_Toc184607955)

[1.2.1. Lớp mô hình họ R-CNN 14](#_Toc184607956)

[1.2.1.1. R-CNN (2014) 14](#_Toc184607957)

[1.2.1.2. Fast R-CNN (2015) 16](#_Toc184607958)

[1.2.1.3. Faster R-CNN (2016) 19](#_Toc184607959)

[1.2.2. Lớp các mô hình họ YOLO 21](#_Toc184607960)

[1.2.2.1 YOLOv1 (2016) 21](#_Toc184607961)

[1.2.2.2. YOLOv2 (2017) 24](#_Toc184607962)

[1.2.2.3. YOLOv3 (2018) 27](#_Toc184607963)

[1.3 Ngôn ngữ lập trình và các thư viện 30](#_Toc184607964)

[1.3.1 Python 30](#_Toc184607965)

[1.3.2. OpenCV (Open Source Computer Vision Library) 31](#_Toc184607966)

[1.3.3. scikit-learn 33](#_Toc184607967)

[1.3.4. NumPy 34](#_Toc184607968)

[1.3.5. Pandas 36](#_Toc184607969)

[CHƯƠNG II: XÂY DỰNG HỆ THỐNG NHẬN DIỆN GIỚI TÍNH VÀ TUỔI SỬ DỤNG OPENCV 38](#_Toc184607970)

[2.1 Yêu cầu của bài toán 38](#_Toc184607971)

[2.2 Các kĩ thuật Deeplearning được dùng trong bài toán mô tả ảnh 40](#_Toc184607972)

[2.2.1. Giai đoạn mã hóa (Encoder) 41](#_Toc184607973)

[2.2.2. Giai đoạn giải mã (Decoder) 41](#_Toc184607974)

[2.2.3. Quy trình tổng thể 42](#_Toc184607975)

[2.2.4. Cải tiến để tăng độ chính xác 43](#_Toc184607976)

[CHƯƠNG III: KẾT QUẢ THỰC NGHIỆM 45](#_Toc184607977)

[3.1 Cơ sở dữ liệu 45](#_Toc184607978)

[3.1.1. Dữ liệu 45](#_Toc184607979)

[3.1.2. Tiền xử lý dữ liệu 45](#_Toc184607980)

[3.2 Độ đo đánh giá 46](#_Toc184607981)

[3.2.1. Độ chính xác (Accuracy): 46](#_Toc184607982)

[3.2.2. Tỷ lệ phát hiện khuôn mặt (Face Detection Rate): 47](#_Toc184607983)

[3.3.3. Độ tin cậy (Confidence Score): 47](#_Toc184607984)

[3.3.4. Tỷ lệ lỗi (Error Rate): 47](#_Toc184607985)

[3.3 Kết quả 48](#_Toc184607986)

[CHƯƠNG IV: KẾT LUẬN 50](#_Toc184607987)

[4.1 Tổng kết 50](#_Toc184607988)

[4.2 Ưu điểm 50](#_Toc184607989)

[4.3 Hạn chế 51](#_Toc184607990)

[4.4 Hướng phát triển tương lai 51](#_Toc184607991)

[4.5 Kết luận 52](#_Toc184607992)

[TÀI LIỆU THAM KHẢO 53](#_Toc184607993)

**DANH MỤC ẢNH**

|  |  |  |
| --- | --- | --- |
| **Số hiệu** | **Tên** | **Trang** |
| 1.1 | Hình minh họa bài toán nhận dạng đối tượng | 9 |
| 1.2 | Ứng dụng trong xe tự lái, với bounding box quanh các phương tiện và người đi bộ | 12 |
| 1.3 | Hình ảnh mô tả kỹ thuật R-CNN | 13 |
| 1.4 | Hình ảnh mô tả kỹ thuật CNN | 16 |
| 1.5 | Hình ảnh mô tả kỹ thuật RNN | 18 |
| 1.6 | Hình ảnh mô tả kỹ thuật YOLO | 21 |
| 1.7 | Hình ảnh mô tả kỹ thuật YOLOv3 | 26 |
| 2.1 | Hình ảnh chưa nhận dạng tuổi và giới tính | 38 |
| 2.2 | Hình ảnh sử lý dữ liệu | 39 |
| 2.3 | Hình ảnh sau khi nhận dạng | 42 |

# LỜI MỞ ĐẦU

Trong thời đại công nghệ phát triển mạnh mẽ như hiện nay, nhận diện khuôn mặt đã trở thành một trong những ứng dụng quan trọng trong nhiều lĩnh vực, bao gồm bảo mật, marketing, y tế và giải trí. Đặc biệt, việc nhận diện giới tính và độ tuổi của một người từ khuôn mặt đã mở ra nhiều cơ hội ứng dụng trong các hệ thống tự động và các dịch vụ cá nhân hóa.

Hệ thống nhận diện giới tính và tuổi thông qua công nghệ nhận diện khuôn mặt có thể cung cấp thông tin quan trọng giúp cải thiện trải nghiệm người dùng, tối ưu hóa các chiến dịch quảng cáo hoặc hỗ trợ trong việc phân tích đối tượng. Trong bối cảnh đó, việc xây dựng và triển khai hệ thống nhận diện giới tính và tuổi là một chủ đề thú vị, vừa có tính ứng dụng cao, vừa đẩy mạnh sự phát triển của trí tuệ nhân tạo (AI) và học máy (Machine Learning) trong thực tế.

Bài tập lớn này trình bày quá trình xây dựng hệ thống nhận diện giới tính và độ tuổi bằng cách sử dụng thư viện OpenCV kết hợp với các mô hình học sâu (deep learning). Chúng tôi sẽ phân tích các bước triển khai hệ thống từ việc tiền xử lý dữ liệu, xây dựng mô hình, đến việc kiểm thử và đánh giá hiệu quả của hệ thống.

Mục tiêu của bài tập lớn là cung cấp cái nhìn tổng quan về cách hệ thống nhận diện giới tính và tuổi hoạt động, đồng thời đánh giá độ chính xác của mô hình qua các thử nghiệm thực tế. Qua đó, hệ thống có thể giúp nâng cao trải nghiệm người dùng và mở ra các ứng dụng trong nhiều lĩnh vực khác nhau.

# CHƯƠNG I: TỔNG QUAN

## 1.1. Nhận dạng đối tượng

### 1.1.1. Định nghĩa nhận dạng đối tượng

Bài toán nhận dạng đối tượng trong thị giác máy tính là quá trình phát hiện, phân loại, và xác định các đối tượng trong hình ảnh hoặc video. Đây là một bài toán cốt lõi của trí tuệ nhân tạo và thị giác máy tính, giúp máy tính "hiểu" và "nhận biết" các đối tượng tương tự như con người.

Ví dụ: Trong một hình ảnh chứa nhiều loại đồ vật khác nhau như ô tô, xe máy, hoặc con người, hệ thống cần phải:

* **Phát hiện đối tượng:** Xác định vị trí và vùng chứa của các đối tượng trong ảnh (bounding box).
* **Phân loại đối tượng:** Gắn nhãn các đối tượng để xác định chúng thuộc loại nào (ví dụ: mèo, chó, xe hơi).
* **Nhận diện đối tượng cụ thể:** Phân biệt các cá thể cụ thể trong cùng một loại (ví dụ: nhận diện khuôn mặt một người).



*Hình 1.1: Hình minh họa bài toán nhận dạng đối tượng*

### 1.1.2. Những vấn đề cần giải quyết trong bài tập nhận dạng

Để giải quyết bài toán nhận dạng đối tượng, cần thực hiện các bước chính sau đây:

1. Thu thập và tiền xử lý dữ liệu:

* Thu thập dữ liệu:
  + Thu thập hình ảnh hoặc video từ nhiều nguồn khác nhau (camera, internet, cơ sở dữ liệu có sẵn).
  + Đảm bảo dữ liệu đa dạng để đại diện cho các tình huống thực tế (góc chụp, môi trường, kích thước đối tượng).
* Tiền xử lý dữ liệu:
  + Chuẩn hóa hình ảnh (kích thước, định dạng).
  + Loại bỏ nhiễu (sử dụng các kỹ thuật như Gaussian Filter, Median Filter).
  + Tăng cường dữ liệu (Data Augmentation): xoay, lật, thay đổi độ sáng để tăng tính đa dạng.

2. Trích xuất đặc trưng:

* Trích xuất đặc trưng thủ công:

Sử dụng các phương pháp như HOG (Histogram of Oriented Gradients), SIFT (Scale-Invariant Feature Transform) để tìm các đặc trưng quan trọng.

* Trích xuất đặc trưng tự động:

Sử dụng mạng nơ-ron tích chập (CNN) để tự động học các đặc trưng phức tạp từ dữ liệu.

3. Phát hiện đối tượng:

* Phát hiện vùng chứa đối tượng:

Sử dụng các thuật toán như Sliding Window, Region Proposal (Selective Search, RPN).

* Các phương pháp hiện đại:

Áp dụng các mô hình nhanh và chính xác như YOLO (You Only Look Once), SSD (Single Shot Detector), Faster R-CNN.

4. Phân loại đối tượng:

* Gắn nhãn cho từng đối tượng:

Phân loại đối tượng vào các lớp cụ thể (ví dụ: mèo, chó, ô tô) bằng các thuật toán học máy hoặc học sâu.

* Mô hình học sâu phổ biến:

ResNet, VGG, InceptionNet, hoặc các mô hình Transformer (Vision Transformer) hiện đại.

5. Hậu xử lý:

* Loại bỏ kết quả không chính xác:

Áp dụng Non-Maximum Suppression (NMS) để chọn bounding box tốt nhất trong trường hợp các kết quả chồng lấp.

* Điều chỉnh kết quả:

Xử lý các lỗi nhỏ để cải thiện độ chính xác và tính trực quan.

6. Đánh giá mô hình:

* Sử dụng các thước đo hiệu suất:
  + Độ chính xác (Accuracy), độ nhạy (Recall), độ đặc hiệu (Precision), F1-score.
  + Mean Average Precision (mAP) là thước đo phổ biến trong nhận dạng đối tượng.
* Kiểm thử trên tập dữ liệu mới:

Đảm bảo mô hình có khả năng tổng quát hóa tốt trên các dữ liệu chưa từng thấy.

7. Tối ưu hóa và triển khai:

* Tối ưu hóa mô hình:

Giảm số lượng tham số hoặc sử dụng mô hình gọn nhẹ (MobileNet, Tiny YOLO) để tăng tốc độ xử lý.

* Triển khai thực tế:

Chạy trên các thiết bị khác nhau như điện thoại, camera giám sát, hoặc thiết bị IoT.

8. Cập nhật và bảo trì:

* Cập nhật dữ liệu:

Thu thập thêm dữ liệu để cải thiện mô hình khi có đối tượng mới hoặc thay đổi môi trường.

* Huấn luyện lại:

Điều chỉnh mô hình khi cần mở rộng thêm lớp đối tượng hoặc nâng cao hiệu suất.

### 1.1.3. Ứng dụng của bài toán nhận dạng đối tượng

Nhận dạng đối tượng có vai trò quan trọng trong nhiều lĩnh vực, bao gồm:

* **Giám sát an ninh:**
  + Nhận diện khuôn mặt trong hệ thống camera an ninh.
  + Phát hiện các hành vi bất thường hoặc đối tượng nguy hiểm.
* **Y tế:**

Phát hiện bệnh lý từ ảnh y khoa, như khối u trong X-quang hoặc CT.

* **Xe tự lái:**

Phát hiện và theo dõi người đi bộ, phương tiện giao thông, biển báo.

* **Thương mại điện tử:**

Tìm kiếm sản phẩm tương tự bằng hình ảnh (Image Search).

* **Giải trí và truyền thông:**

Ứng dụng trong nhận diện biểu cảm khuôn mặt hoặc hiệu ứng thực tế ảo (AR).



*Hình 1.2: Ứng dụng trong xe tự lái, với bounding box quanh các phương tiện và người đi bộ*

## 1.2 Các phương pháp áp dụng cho bài toán nhận dạng

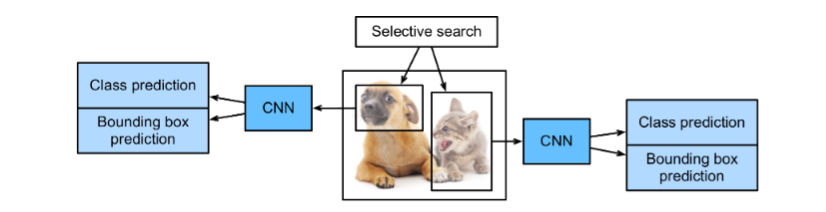
### 1.2.1. Lớp mô hình họ R-CNN

#### 1.2.1.1. R-CNN (2014)

R-CNN là một mô hình hai giai đoạn. Trong giai đoạn đầu, Selective Search được sử dụng để tạo ra các vùng đề xuất (region proposals) trong ảnh. Các vùng này có thể chứa các đối tượng mà mô hình cần phát hiện. Sau đó, mỗi vùng đề xuất được cắt ra và đưa qua một mạng Convolutional Neural Network (CNN) đã được huấn luyện để trích xuất các đặc trưng. Những đặc trưng này sau đó được sử dụng để phân loại đối tượng và xác định vị trí của các hộp giới hạn (bounding boxes). Việc áp dụng CNN giúp R-CNN tự động học các đặc trưng từ ảnh, thay vì phải sử dụng các đặc trưng thủ công như các phương pháp trước đây (SIFT, HOG, v.v.) **[1]**.

**Kiến trúc mô hình**

R-CNN sử dụng ba bước chính trong quá trình phát hiện đối tượng:

****

*Hình 1.3: Hình ảnh mô tả kỹ thuật R-CNN*

1. **Selective Search:** R-CNN sử dụng Selective Search để tìm ra các vùng đối tượng tiềm năng trong ảnh. Phương pháp này phân vùng ảnh thành các vùng nhỏ hơn và kết hợp chúng lại, nhằm xác định các đối tượng có thể xuất hiện trong ảnh. Selective Search không yêu cầu thông tin bổ sung về đối tượng trong ảnh, chỉ cần phân tích hình ảnh để tạo ra hàng nghìn vùng đề xuất tiềm năng [1].
2. **Trích xuất đặc trưng bằng CNN:** Sau khi các vùng đề xuất được tạo ra, mỗi vùng sẽ được đưa vào một mạng CNN để trích xuất các đặc trưng. Các đặc trưng này được sử dụng cho việc phân loại đối tượng và xác định vị trí của các hộp giới hạn. CNN đã được huấn luyện trên một tập dữ liệu lớn (như ImageNet) để học các đặc trưng ảnh mà không cần phải sử dụng các đặc trưng thủ công [1].
3. **Phân loại và Hồi quy:** Sau khi các đặc trưng được trích xuất, mỗi vùng được phân loại thành các đối tượng khác nhau bằng cách sử dụng một bộ phân loại SVM (Support Vector Machine). Đồng thời, một mô hình hồi quy được sử dụng để điều chỉnh vị trí của các hộp giới hạn, giúp xác định chính xác hơn vị trí của đối tượng trong ảnh [1].

**Ưu điểm:**

* **Độ chính xác cao:** R-CNN đã đạt được một sự cải thiện đáng kể về độ chính xác trong phát hiện đối tượng so với các phương pháp trước đó, đặc biệt là khi so sánh với các mô hình như DPM (Deformable Part Models) và HOG (Histogram of Oriented Gradients) [1].
* **Sử dụng CNN để trích xuất đặc trưng:** Việc sử dụng CNN trong R-CNN giúp mô hình tự động học các đặc trưng từ ảnh, điều này đã cải thiện hiệu suất so với các phương pháp truyền thống cần đặc trưng thủ công [1].
* **Khả năng phát hiện đối tượng trong ảnh:** R-CNN có khả năng phát hiện các đối tượng trong ảnh với độ chính xác cao, nhờ vào sự kết hợp của Selective Search và CNN. Nó có thể nhận diện các đối tượng có sự thay đổi về kích thước, hình dạng và vị trí trong ảnh [1].

**Nhược điểm:**

* **Tốc độ chậm:** Một trong những nhược điểm lớn của R-CNN là tốc độ tính toán. Việc xử lý từng vùng đề xuất qua CNN và sử dụng Selective Search tạo ra chi phí tính toán cao, dẫn đến thời gian xử lý chậm, đặc biệt là đối với các ảnh lớn hoặc video [1].
* **Chi phí tính toán cao:** R-CNN cần tính toán rất nhiều, vì mỗi vùng đề xuất phải được xử lý qua CNN một cách riêng biệt, dẫn đến việc tốn bộ nhớ và tài nguyên tính toán [1].
* **Không tối ưu hóa đầu cuối:** Mô hình R-CNN không thể tối ưu hóa toàn bộ quy trình phát hiện đối tượng, vì các giai đoạn trích xuất đặc trưng, phân loại và hồi quy được huấn luyện độc lập, không tối ưu hóa đồng thời cho tất cả các nhiệm vụ [1].

**Ứng dụng**

R-CNN đã được áp dụng thành công trong nhiều bài toán nhận diện đối tượng trong các lĩnh vực khác nhau:

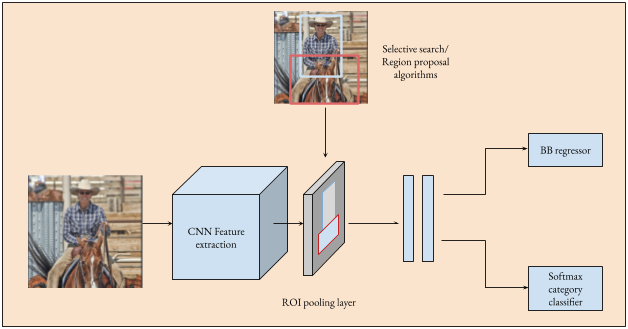
* **Phát hiện đối tượng trong ảnh:** R-CNN có thể phát hiện các đối tượng trong ảnh, từ người, xe, động vật cho đến các vật thể phức tạp khác [1].
* **Xe tự lái:** Mô hình này có thể được sử dụng trong các hệ thống nhận dạng vật thể cho các xe tự lái, giúp phát hiện và nhận dạng các đối tượng trên đường [1].
* **Giám sát video:** R-CNN có thể được sử dụng trong các hệ thống giám sát video để phát hiện các hành vi bất thường hoặc các đối tượng trong cảnh quay [1].

#### 1.2.1.2. Fast R-CNN (2015)

Fast R-CNN được thiết kế để cải thiện tốc độ và độ chính xác của quá trình phát hiện đối tượng. Thay vì chạy mạng CNN trên từng vùng ứng viên (region proposal), Fast R-CNN chỉ chạy một lần trên toàn bộ ảnh để tạo ra bản đồ đặc trưng (feature map). Từ đó, các vùng ứng viên được ánh xạ và xử lý dựa trên bản đồ đặc trưng này, giúp giảm đáng kể thời gian tính toán mà vẫn giữ được hiệu suất cao [2].

**Kiến trúc mô hình:**

Kiến trúc của Fast R-CNN bao gồm các bước chính sau:



*Hình 1.4: Hình ảnh mô tả kỹ thuật CNN*

1. **Trích xuất đặc trưng toàn ảnh:** Ảnh đầu vào được đưa qua một mạng CNN (thường là VGG-16 hoặc ResNet) để tạo bản đồ đặc trưng. Đây là bước tính toán quan trọng, được thực hiện chỉ một lần trên toàn bộ ảnh [2].
2. **Region of Interest (RoI) Pooling:** Các vùng ứng viên (Region Proposals), được tạo bởi Selective Search, được ánh xạ lên bản đồ đặc trưng. Sau đó, lớp RoI Pooling chuẩn hóa kích thước của các đặc trưng từ mỗi vùng ứng viên để phù hợp với các bước tiếp theo [2].
3. **Fully Connected Layer:** Đặc trưng từ RoI Pooling được đưa qua một mạng fully connected để thực hiện hai nhiệm vụ song song:

* **Phân loại:** Dự đoán lớp của đối tượng.
* **Hồi quy hộp giới hạn:** Tinh chỉnh tọa độ hộp giới hạn để cải thiện độ chính xác [2].

1. **Học sâu đầu-cuối (End-to-End Learning):** Mô hình này kết hợp phân loại và hồi quy trong một mạng duy nhất, giúp tối ưu hóa hiệu suất một cách toàn diện [2][3].

**Ưu điểm**

* **Tốc độ xử lý nhanh hơn:** Thay vì chạy mạng CNN cho từng vùng ứng viên, Fast R-CNN chỉ cần chạy CNN một lần trên toàn bộ ảnh, giúp tăng tốc quá trình phát hiện đối tượng [2].
* **Học sâu đầu-cuối (End-to-End Learning):** Việc tối ưu hóa đồng thời cả hai nhiệm vụ (phân loại và hồi quy) giúp mô hình học được các đặc trưng tốt hơn, từ đó cải thiện hiệu suất [3].
* **Sử dụng ít tài nguyên hơn:** Fast R-CNN chỉ cần lưu một bản đồ đặc trưng cho toàn ảnh thay vì lưu đặc trưng riêng cho từng vùng ứng viên, giúp giảm yêu cầu bộ nhớ và tài nguyên tính toán [1]. Việc sử dụng mạng CNN sâu như VGG hoặc ResNet cũng giúp nâng cao khả năng trích xuất đặc trưng mạnh mẽ từ ảnh [4].

**Nhược điểm**

* **Phụ thuộc vào Selective Search:** Mặc dù nhanh hơn R-CNN, Fast R-CNN vẫn sử dụng Selective Search để tạo các vùng ứng viên, một bước chậm và không tối ưu [2].
* **Không đạt thời gian thực:** Dù cải tiến đáng kể, Fast R-CNN vẫn không đủ nhanh để áp dụng cho các ứng dụng yêu cầu thời gian thực, như xe tự hành hoặc nhận diện trong video trực tiếp [3].

**Ứng dụng**

Fast R-CNN có nhiều ứng dụng quan trọng trong thực tế:

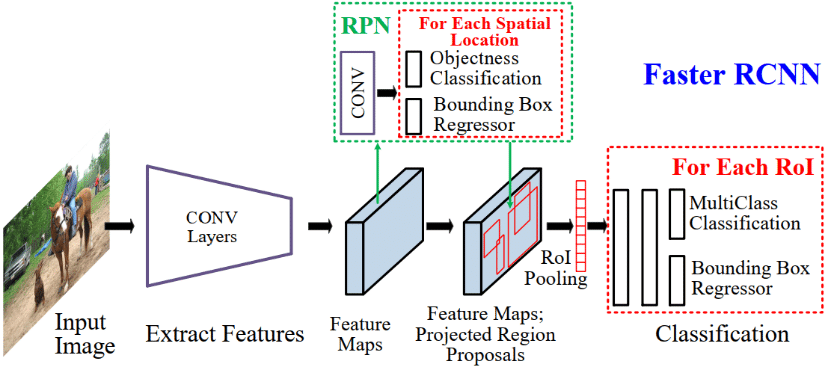
* **Phát hiện đối tượng trong ảnh và video:** Sử dụng trong giám sát giao thông, xe tự hành, hoặc nhận dạng khuôn mặt.
* **Y tế:** Phát hiện khối u hoặc các bất thường trong ảnh y khoa (MRI, X-ray).
* **Công nghiệp:** Phân tích và kiểm tra sản phẩm trên dây chuyền sản xuất tự động [2].

#### 1.2.1.3. Faster R-CNN (2016)

Faster R-CNN cải thiện Fast R-CNN bằng cách tích hợp một mạng đề xuất vùng (RPN) vào quá trình phát hiện đối tượng. Thay vì sử dụng Selective Search như trong Fast R-CNN để tạo các vùng ứng viên, RPN học tự động các vùng ứng viên tiềm năng thông qua mạng nơ-ron. Điều này giúp giảm thiểu độ trễ và tối ưu hóa toàn bộ quá trình phát hiện đối tượng trong một mô hình học sâu duy nhất [3].

**Kiến trúc của mô hình:**

Kiến trúc của Faster R-CNN bao gồm ba phần chính:



*Hình 1.5: Hình ảnh mô tả kỹ thuật RNN*

1. **Mạng CNN (**Backbone Network**):** Giống như trong Fast R-CNN, Faster R-CNN sử dụng một mạng CNN (ví dụ: VGG-16, ResNet) để trích xuất đặc trưng từ ảnh đầu vào. Bước này tạo ra một bản đồ đặc trưng (feature map) cho toàn bộ ảnh [3].
2. **Region Proposal Network (RPN):**

* RPN là một mạng con được tích hợp trong Faster R-CNN và giúp tự động sinh các vùng ứng viên. Nó sử dụng các cửa sổ trượt trên bản đồ đặc trưng và mỗi cửa sổ này sẽ dự đoán xem có một đối tượng nào trong khu vực đó không, đồng thời tinh chỉnh vị trí của các hộp giới hạn (bounding boxes). Điều này thay thế Selective Search, giúp Faster R-CNN nhanh hơn nhiều [3].
* RPN sử dụng một lớp phân loại nhị phân để xác định liệu có đối tượng trong một cửa sổ hay không, và một lớp hồi quy để điều chỉnh tọa độ của các hộp giới hạn.

1. **RoI Pooling:** Tương tự như Fast R-CNN, Faster R-CNN sử dụng RoI Pooling để chuẩn hóa kích thước của các vùng ứng viên. Các đặc trưng của các vùng ứng viên sau đó được đưa qua một mạng fully connected để phân loại đối tượng và hồi quy hộp giới hạn.
2. **Đào tạo end-to-end:** Toàn bộ mô hình, bao gồm mạng CNN và RPN, có thể được huấn luyện đồng thời trong một quá trình học sâu end-to-end, giúp tối ưu hóa toàn bộ hệ thống phát hiện đối tượng.

**Ưu điểm**

* **Hiệu suất nhanh hơn và chính xác hơn:** Thay thế Selective Search bằng RPN giúp Faster R-CNN nhanh hơn rất nhiều, đồng thời duy trì hoặc cải thiện độ chính xác phát hiện đối tượng. Quá trình tạo vùng ứng viên được tối ưu hóa, và mô hình có thể được huấn luyện end-to-end mà không cần phải tách biệt giữa các bước [3].
* **Độ chính xác cao:** Nhờ khả năng tự động tạo vùng ứng viên với RPN, Faster R-CNN đạt được độ chính xác cao hơn so với các phương pháp trước đó như R-CNN và Fast R-CNN [5].
* **Ứng dụng rộng rãi:** Faster R-CNN có thể áp dụng cho nhiều lĩnh vực khác nhau, từ giám sát an ninh đến nhận dạng đối tượng trong các video, và trong các ứng dụng y tế hoặc công nghiệp [3].

**Nhược điểm**

* **Vẫn chưa đủ nhanh cho các ứng dụng thời gian thực:** Mặc dù Faster R-CNN nhanh hơn nhiều so với R-CNN và Fast R-CNN, mô hình này vẫn chưa đạt được tốc độ đủ nhanh cho các ứng dụng thời gian thực, như xe tự lái hoặc nhận dạng đối tượng trong video trực tiếp [5].
* **Khó khăn trong việc xử lý các đối tượng nhỏ:** Mặc dù Faster R-CNN đạt được kết quả tuyệt vời trong hầu hết các tác vụ, nó vẫn gặp khó khăn trong việc phát hiện các đối tượng nhỏ hoặc đối tượng có tỉ lệ biến dạng lớn, vì RPN có thể không tạo ra các vùng ứng viên đủ chính xác cho những đối tượng này [3].

**Ứng dụng**

Faster R-CNN có nhiều ứng dụng quan trọng trong nhiều lĩnh vực, bao gồm:

* **Phát hiện đối tượng trong ảnh và video:** Faster R-CNN đã được sử dụng trong các hệ thống giám sát video, nhận dạng khuôn mặt, phát hiện đối tượng trong giao thông, v.v.
* **Ứng dụng y tế:** Phát hiện tổn thương trong các ảnh y khoa như MRI, CT, X-ray, giúp bác sĩ chẩn đoán nhanh chóng và chính xác hơn.
* **Công nghiệp và sản xuất:** Phát hiện lỗi trong quá trình sản xuất và giám sát dây chuyền sản xuất tự động [3].

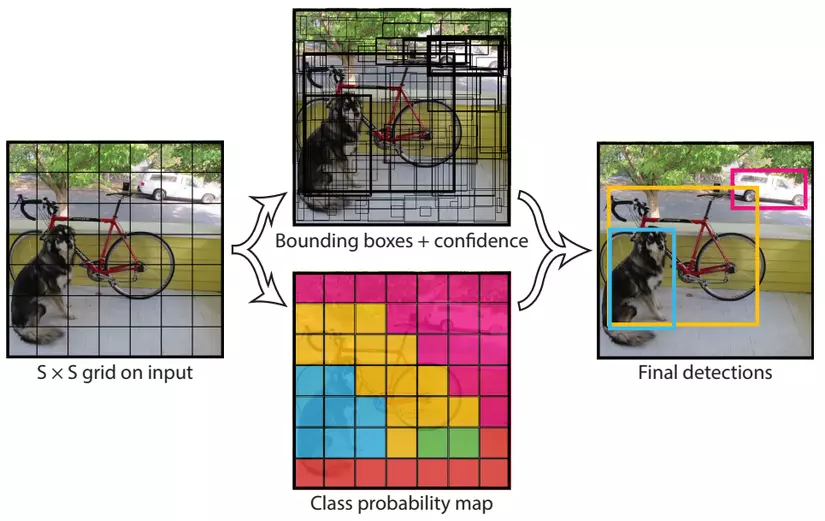
### 1.2.2. Lớp các mô hình họ YOLO

#### 1.2.2.1 YOLOv1 (2016)

Khái niệm cơ bản của YOLOv1 là thực hiện phát hiện đối tượng qua một lần duy nhất, thay vì chia quá trình thành nhiều bước như các phương pháp trước đây. Cụ thể, mô hình YOLOv1 áp dụng mạng CNN (Convolutional Neural Network) để phân tích toàn bộ ảnh đầu vào và trực tiếp dự đoán các hộp giới hạn (bounding boxes) cùng với các xác suất phân loại cho từng lớp đối tượng trong một bước duy nhất. Điều này có nghĩa là YOLOv1 phân tích ảnh đầu vào toàn bộ trong một lần duy nhất, thay vì phải tìm kiếm từng vùng ứng viên như trong các mô hình trước đây (R-CNN, Fast R-CNN) **[5]**.

**Kiến trúc của mô hình:**

YOLOv1 sử dụng một mạng CNN duy nhất với kiến trúc gồm nhiều lớp để xử lý ảnh đầu vào. Các bước chính trong kiến trúc của YOLOv1 như sau:



*Hình 1.6: Hình ảnh mô tả kỹ thuật YOLO*

1. **Input Image:** YOLOv1 nhận vào một ảnh có kích thước cố định (thường là 224x224 pixels hoặc 448x448 pixels). Mô hình sẽ chia ảnh thành một lưới (grid) có kích thước cố định (ví dụ: 7x7).
2. **Mạng CNN:** Mạng CNN được thiết kế để trích xuất các đặc trưng của ảnh đầu vào. Mạng này có nhiều lớp convolutional và max pooling để giảm thiểu kích thước của ảnh trong khi vẫn giữ lại các đặc trưng quan trọng. Các đặc trưng này sau đó được sử dụng để dự đoán các hộp giới hạn và phân loại đối tượng.
3. **Lưới phân chia ảnh:** Ảnh đầu vào được chia thành một lưới có kích thước S x S (ví dụ: 7 x 7). Mỗi ô trong lưới này sẽ dự đoán một số lượng hộp giới hạn và xác suất cho các đối tượng trong ô đó.
4. **Bounding Box Predictions:** Mỗi ô trong lưới dự đoán một số lượng hộp giới hạn (khoảng 2 hộp giới hạn) và vị trí của chúng, bao gồm 4 giá trị: tọa độ (x, y) của tâm hộp, chiều rộng (w) và chiều cao (h) của hộp. Các giá trị này được chuẩn hóa để đảm bảo sự linh hoạt với các ảnh có kích thước khác nhau.
5. **Class Probability Predictions:** Mỗi ô trong lưới còn dự đoán xác suất cho các lớp đối tượng khác nhau. Các lớp này được xác định dựa trên bài toán phân loại đối tượng.
6. **Output:** Kết quả đầu ra của YOLOv1 là một vector có kích thước cố định, chứa các dự đoán về các hộp giới hạn và các xác suất phân loại cho mỗi ô lưới [5].

**Ưu điểm:**

* **Tốc độ phát hiện nhanh:** Một trong những điểm mạnh lớn nhất của YOLOv1 là khả năng phát hiện đối tượng trong thời gian thực. Mô hình này có thể xử lý hàng nghìn ảnh mỗi giây, điều này làm cho YOLO trở thành lựa chọn lý tưởng cho các ứng dụng thời gian thực như giám sát video hoặc xe tự lái.
* **Phát hiện đối tượng trong toàn bộ ảnh:** Khác với các phương pháp như R-CNN, YOLOv1 không cần phải phân chia ảnh thành các vùng ứng viên. Thay vào đó, YOLOv1 phân tích toàn bộ ảnh trong một lần, giúp giảm độ phức tạp và tăng tốc độ.
* **Chính xác trong việc phát hiện các đối tượng lớn:** YOLOv1 hoạt động rất tốt khi phát hiện các đối tượng có kích thước lớn, với độ chính xác cao khi đối tượng chiếm một phần lớn trong ảnh [5].

**Nhược điểm:**

* **Khó phát hiện các đối tượng nhỏ:** Vì ảnh được chia thành các ô lưới và mỗi ô chỉ dự đoán một hoặc hai hộp giới hạn, YOLOv1 gặp khó khăn trong việc phát hiện các đối tượng nhỏ hoặc các đối tượng ở góc ảnh [5].
* **Độ chính xác thấp khi có đối tượng chồng lấn:** YOLOv1 không thể phát hiện chính xác khi có nhiều đối tượng gần nhau hoặc chồng lấn trong cùng một ô lưới [5].
* **Chất lượng phân loại thấp:** Do cách tiếp cận này chỉ dự đoán một số ít lớp cho mỗi ô, chất lượng phân loại trong YOLOv1 thấp hơn so với các phương pháp khác như R-CNN hoặc Fast R-CNN [5].

**Ứng dụng**

YOLOv1 đã được áp dụng trong nhiều ứng dụng thực tế, đặc biệt là trong các hệ thống yêu cầu thời gian thực, bao gồm:

* **Giám sát video:** YOLOv1 có thể phát hiện và theo dõi đối tượng trong video với tốc độ cao, rất hữu ích trong các ứng dụng an ninh và giám sát.
* **Xe tự lái:** YOLOv1 được sử dụng trong các hệ thống xe tự lái để phát hiện các vật cản và các đối tượng xung quanh xe, giúp đưa ra các quyết định lái xe tự động.
* **Nhận diện đối tượng trong các ứng dụng di động:** Với khả năng xử lý nhanh, YOLOv1 cũng được triển khai trong các ứng dụng di động để phát hiện và phân loại các đối tượng trong ảnh [5].

#### 1.2.2.2. YOLOv2 (2017)

YOLOv2 giữ nguyên ý tưởng cơ bản của YOLOv1 là sử dụng mạng CNN để dự đoán các hộp giới hạn (bounding boxes) và các xác suất phân loại đối tượng trong một lần duy nhất. Tuy nhiên, YOLOv2 cải thiện nhiều yếu tố về kiến trúc và chiến lược huấn luyện, chẳng hạn như sử dụng kỹ thuật tối ưu hóa mới và hệ thống dữ liệu đa lớp. Mô hình YOLOv2 cũng hỗ trợ việc phát hiện nhiều đối tượng trong một ảnh một cách chính xác hơn so với YOLOv1. Đặc biệt, YOLOv2 sử dụng các kỹ thuật như batch normalization và anchor boxes để nâng cao khả năng học các đặc trưng từ dữ liệu huấn luyện **[6]**.

**Kiến trúc của mô hình:**

**YOLOv2** có nhiều cải tiến so với YOLOv1, bao gồm các điểm nổi bật sau:

1. **Sử dụng Darknet-19**: YOLOv2 sử dụng một kiến trúc mạng CNN mới gọi là Darknet-19 thay thế mạng CNN trong YOLOv1. Darknet-19 có 19 lớp, kết hợp giữa các lớp convolutional và max pooling. Đặc biệt, Darknet-19 nhẹ hơn và hiệu quả hơn so với mạng CNN mà YOLOv1 sử dụng. Điều này giúp giảm số lượng tham số và tăng tốc độ huấn luyện.
2. **Batch Normalization**: YOLOv2 áp dụng batch normalization vào các lớp convolutional, giúp làm cho quá trình huấn luyện nhanh chóng hơn và ổn định hơn. Batch normalization giúp giảm thiểu hiện tượng vanishing gradients và cải thiện khả năng tổng quát của mô hình, giúp mô hình học được các đặc trưng mạnh mẽ hơn từ dữ liệu huấn luyện.
3. **Anchor Boxes**: Một cải tiến quan trọng trong YOLOv2 là việc sử dụng anchor boxes (hộp giới hạn tham chiếu) để cải thiện khả năng dự đoán kích thước và tỉ lệ của các đối tượng. Thay vì chỉ dự đoán một hoặc hai hộp giới hạn cho mỗi ô trong lưới, YOLOv2 dự đoán nhiều hộp giới hạn (anchor boxes) cho mỗi ô lưới, giúp cải thiện độ chính xác trong việc phát hiện đối tượng có kích thước và hình dạng khác nhau.
4. **Cải thiện Lưới (Grid) và Kích thước Ảnh**: YOLOv2 có thể sử dụng các ảnh đầu vào với kích thước lớn hơn và lưới phân chia ảnh cũng được cải tiến. Điều này giúp mô hình có thể nhận diện đối tượng nhỏ hơn một cách chính xác hơn, cải thiện khả năng phát hiện đối tượng trong các cảnh phức tạp hơn.
5. **Multi-Scale Training**: YOLOv2 giới **thiệu huấn luyện đa tỉ lệ (multi**-scale training), tức là trong quá trình huấn luyện, ảnh đầu vào sẽ được thay đổi kích thước ngẫu nhiên để giúp mô hình có thể nhận diện các đối tượng ở nhiều tỉ lệ khác nhau, từ đó cải thiện khả năng phát hiện đối tượng nhỏ.

**Ưu điểm:**

* **Cải thiện độ chính xác**: So với YOLOv1, YOLOv2 cải thiện đáng kể độ chính xác trong việc phát hiện đối tượng, đặc biệt là đối với các đối tượng nhỏ và những đối tượng có tỷ lệ khác nhau.
* **Tốc độ nhanh**: Dù có nhiều cải tiến về kiến trúc, YOLOv2 vẫn duy trì được tốc độ phát hiện rất nhanh, giúp xử lý video thời gian thực và các ứng dụng yêu cầu độ trễ thấp.
* **Khả năng phát hiện đa lớp**: YOLOv2 có thể nhận diện một lượng lớn các đối tượng khác nhau, lên đến **9000 lớp đối tượng**, nhờ vào việc cải tiến mô hình học của nó. Điều này giúp YOLOv2 có thể xử lý các bài toán phát hiện đối tượng với nhiều lớp hơn so với YOLOv1.
* **Hỗ trợ phát hiện đối tượng ở nhiều kích thước khác nhau**: Với cải tiến anchor boxes và multi-scale training, YOLOv2 có khả năng phát hiện đối tượng ở nhiều kích thước khác nhau, giúp tăng cường khả năng nhận diện trong các tình huống thực tế phức tạp.

**Nhược điểm:**

* **Khó phát hiện đối tượng quá nhỏ**: Mặc dù YOLOv2 đã cải thiện khả năng phát hiện đối tượng nhỏ hơn so với YOLOv1, nhưng vẫn có thể gặp khó khăn khi phát hiện các đối tượng có kích thước cực kỳ nhỏ trong ảnh.
* **Chất lượng phân loại chưa cao khi đối tượng chồng lấn**: Mặc dù có thể phát hiện nhiều đối tượng trong cùng một ô lưới, nhưng khi các đối tượng chồng lấn nhau hoặc quá gần nhau, YOLOv2 vẫn có thể gặp phải vấn đề trong việc phân loại chính xác.

**Ứng dụng**

YOLOv2 được áp dụng rộng rãi trong nhiều lĩnh vực, đặc biệt là các ứng dụng yêu cầu phát hiện đối tượng nhanh và chính xác. Một số ứng dụng điển hình của YOLOv2 bao gồm:

* **Giám sát video**: YOLOv2 có thể xử lý video thời gian thực và phát hiện các đối tượng trong môi trường giám sát, giúp phát hiện các hành vi bất thường.
* **Hệ thống xe tự lái**: YOLOv2 được sử dụng trong các hệ thống xe tự lái để phát hiện các vật thể như người đi bộ, xe cộ, biển báo giao thông, giúp xe đưa ra quyết định điều khiển.
* **Ứng dụng robot và tự động hóa**: YOLOv2 được triển khai trong các ứng dụng robot để phát hiện các đối tượng và giúp robot tương tác với môi trường một cách hiệu quả.
* **Phát hiện trong các ứng dụng di động**: YOLOv2 cũng được áp dụng trong các ứng dụng di động, như nhận diện vật thể trong các bức ảnh hoặc video, giúp tăng trải nghiệm người dùng.

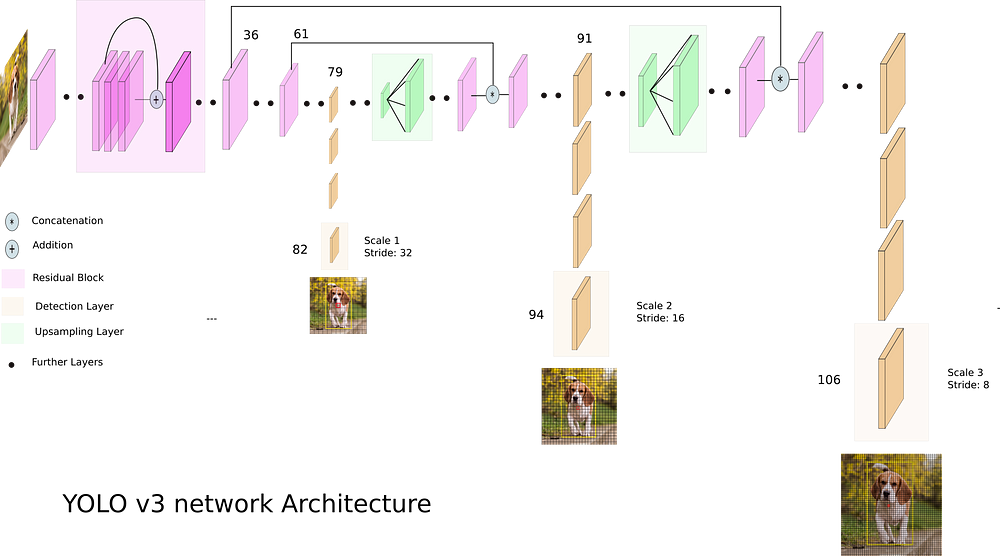
#### 1.2.2.3. YOLOv3 (2018)

YOLOv3 tiếp tục sử dụng ý tưởng cơ bản của YOLO, nhưng với những cải tiến quan trọng. Mô hình này sử dụng một mạng CNN sâu hơn và mạnh mẽ hơn để cải thiện khả năng phát hiện đối tượng ở mọi kích thước. Một trong những điểm đặc biệt của YOLOv3 là việc áp dụng tầng dự đoán đa cấp (multi-scale prediction), cho phép mô hình phát hiện đối tượng ở nhiều kích thước khác nhau trong các tầng mạng sâu hơn.

YOLOv3 sử dụng Darknet-53, một mạng CNN sâu hơn, thay cho Darknet-19 của YOLOv2. Điều này giúp cải thiện khả năng học và xử lý các đặc trưng phức tạp từ ảnh, đặc biệt trong các tình huống đối tượng có hình dạng phức tạp và nền đa dạng. YOLOv3 không chỉ cải thiện độ chính xác mà còn giữ được tốc độ nhanh, giúp mô hình có thể sử dụng trong các ứng dụng thời gian thực **[7]**.

**Kiến trúc của mô hình:**

YOLOv3 có một số cải tiến đáng chú ý so với các phiên bản trước:



*Hình 1.7*: *Hình ảnh mô tả kỹ thuật YOLO*

1. **Sử dụng Darknet-53**: YOLOv3 sử dụng Darknet-53, một mạng CNN với 53 lớp, thay thế cho Darknet-19 trong YOLOv2. Darknet-53 mạnh mẽ hơn, giúp mạng học được các đặc trưng phức tạp hơn và cải thiện độ chính xác khi phát hiện đối tượng. Mạng này sử dụng một loạt các lớp convolutional và residual connections, giúp cải thiện việc huấn luyện và giảm thiểu vấn đề vanishing gradients**.**
2. **Multi-Scale Prediction**: Một cải tiến quan trọng trong YOLOv3 là việc sử dụng multi-scale prediction, tức là mô hình không chỉ dự đoán các đối tượng từ một tầng duy nhất mà từ nhiều tầng khác nhau. Điều này cho phép YOLOv3 phát hiện đối tượng ở nhiều kích thước khác nhau trong ảnh, đặc biệt là các đối tượng nhỏ và lớn.
3. **Residual Connections**: YOLOv3 sử **dụng residual connections**, một kỹ thuật giúp cải thiện khả năng huấn luyện và độ chính xác của mô hình. Các residual connections cho phép mạng học được các đặc trưng phức tạp mà không gặp phải vấn đề của các mạng sâu, như vanishing gradients.
4. **Kỹ thuật Class Prediction và Bounding Box Prediction**: YOLOv3 cải thiện cách dự đoán phân loại đối tượng và các hộp giới hạn. Mô hình này sử dụng logistic regression cho việc dự đoán xác suất phân loại và dự đoán tầng tự động cho các bounding box, giúp cải thiện độ chính xác trong việc phát hiện đối tượng.
5. **Cải thiện Anchor Boxes**: YOLOv3 sử dụng một chiến lược anchor box mới để phát hiện đối tượng tốt hơn. Các anchor box này giúp mô hình dự đoán chính xác hơn kích thước và vị trí của các đối tượng trong ảnh. Việc cải tiến số lượng và tỉ lệ của các anchor boxes giúp mô hình phát hiện đối tượng với tỷ lệ khác nhau **[7]**.

**Ưu điểm:**

* **Phát hiện đối tượng chính xác**: YOLOv3 cải thiện độ chính xác trong việc phát hiện đối tượng, đặc biệt là đối với các đối tượng nhỏ và có sự thay đổi về tỷ lệ. Multi-scale prediction giúp mô hình có khả năng phát hiện đối tượng ở nhiều kích thước khác nhau.
* **Tốc độ nhanh**: Mặc dù có sự cải tiến về độ chính xác, YOLOv3 vẫn duy trì được tốc độ phát hiện nhanh chóng, giúp xử lý video và các ứng dụng thời gian thực hiệu quả.
* **Khả năng phát hiện đối tượng tốt hơn**: Mô hình có khả năng phát hiện một lượng lớn đối tượng trong một ảnh với độ chính xác cao nhờ vào việc sử dụng các anchor boxes và multi-scale prediction.
* **Sử dụng trong nhiều ứng dụng thực tế**: YOLOv3 có thể áp dụng cho nhiều ứng dụng thực tế như giám sát video, nhận diện vật thể trong các ảnh và video, và cả trong các hệ thống xe tự lái **[7]**.

**Nhược điểm:**

* **Vẫn có vấn đề với các đối tượng nhỏ**: Dù đã có cải tiến với multi-scale prediction, YOLOv3 vẫn gặp khó khăn trong việc phát hiện các đối tượng quá nhỏ, đặc biệt là trong những ảnh có nền phức tạp.
* **Độ chính xác của phân loại có thể bị ảnh hưởng khi có nhiều đối tượng chồng lấn**: Mặc dù khả năng phát hiện đối tượng của YOLOv3 rất tốt, nhưng trong một số trường hợp khi có nhiều đối tượng chồng lấn nhau, mô hình có thể gặp khó khăn trong việc phân loại chính xác **[7]**.

**5. Ứng dụng**

YOLOv3 được sử dụng rộng rãi trong nhiều lĩnh vực nhờ vào khả năng phát hiện đối tượng nhanh và chính xác. Các ứng dụng phổ biến của YOLOv3 bao gồm:

* **Giám sát video và an ninh**: YOLOv3 được sử dụng trong các hệ thống giám sát video để phát hiện các hành vi bất thường, như nhận diện người, xe cộ, và các đối tượng khác trong môi trường giám sát.
* **Xe tự lái**: YOLOv3 đóng vai trò quan trọng trong các hệ thống xe tự lái, giúp phát hiện các vật thể xung quanh, bao gồm người đi bộ, xe cộ và các vật cản, giúp xe đưa ra các quyết định điều khiển.
* **Ứng dụng trong robot**: YOLOv3 cũng được áp dụng trong các ứng dụng robot, như nhận diện đối tượng và điều khiển robot thực hiện các tác vụ trong môi trường thực tế.
* **Nhận diện vật thể trong ảnh và video**: YOLOv3 được sử dụng trong nhiều ứng dụng nhận diện vật thể trong các ảnh và video, như trong các ứng dụng di động, dịch vụ tìm kiếm hình ảnh và video, và phân tích thị trường **[7]**.

## 1.3 Ngôn ngữ lập trình và các thư viện

### 1.3.1 Python

Python là một ngôn ngữ lập trình rất phổ biến trong các lĩnh vực học máy, học sâu và thị giác máy tính. Sở dĩ Python được ưa chuộng là vì:

* **Cú pháp đơn giản và dễ đọc:** Python giúp các nhà nghiên cứu và lập trình viên dễ dàng viết mã và hiểu các mô hình phức tạp mà không bị rối loạn trong cú pháp.
* **Thư viện phong phú:** Python có một hệ sinh thái rất mạnh mẽ với các thư viện hỗ trợ cho nhận dạng đối tượng và thị giác máy tính, giúp giảm thiểu đáng kể thời gian phát triển các mô hình.

Các thư viện Python quan trọng trong nhận dạng đối tượng:

1. **TensorFlow và Keras:**
   * TensorFlow là thư viện mã nguồn mở mạnh mẽ của Google, được sử dụng chủ yếu cho các ứng dụng học sâu. Keras là API cấp cao cho phép xây dựng mô hình học sâu một cách dễ dàng trên TensorFlow.
   * Ứng dụng: Nhận dạng đối tượng bằng CNN, YOLO, Faster R-CNN.
2. **PyTorch:**
   * PyTorch là một thư viện học sâu mã nguồn mở được phát triển bởi Facebook. PyTorch nổi bật với khả năng tính toán động và dễ dàng triển khai các mô hình học sâu, đặc biệt trong lĩnh vực thị giác máy tính.
   * Ứng dụng: Xây dựng và huấn luyện mô hình CNN, RNN, YOLO, và các mô hình học sâu khác.
3. **scikit-learn:**
   * Thư viện học máy phổ biến, cung cấp các thuật toán phân loại, hồi quy và các công cụ xử lý dữ liệu. Dù không chuyên sâu về học sâu, scikit-learn vẫn rất hữu ích trong các bài toán nhận dạng đối tượng khi kết hợp với các mô hình học máy như SVM.
   * Ứng dụng: Phân loại đối tượng, giảm chiều dữ liệu, chọn lựa đặc trưng.
4. **OpenCV:**
   * OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở mạnh mẽ cho các ứng dụng thị giác máy tính và xử lý ảnh.
   * Ứng dụng: Tiền xử lý hình ảnh, phát hiện đối tượng, nhận diện khuôn mặt, video tracking, xử lý ảnh trong thời gian thực.
5. **NumPy và Pandas:**
   * NumPy là thư viện hỗ trợ mảng (arrays) và các phép toán số học mạnh mẽ, trong khi Pandas là thư viện giúp xử lý và phân tích dữ liệu, đặc biệt là dữ liệu bảng.
   * Ứng dụng: Xử lý và quản lý dữ liệu ảnh và đặc trưng trước khi đưa vào mô hình học máy.

**Ưu điểm khi sử dụng Python:**

* Được hỗ trợ bởi cộng đồng lập trình viên lớn và các tài liệu hướng dẫn phong phú.
* Dễ dàng tích hợp với các thư viện khác nhau trong hệ sinh thái AI, giúp xây dựng và triển khai mô hình nhanh chóng.

### 1.3.2. OpenCV (Open Source Computer Vision Library)

OpenCV là một thư viện mã nguồn mở rất mạnh mẽ trong việc xử lý ảnh và video. Được phát triển bởi Intel, OpenCV cung cấp một loạt các công cụ và thuật toán giúp thực hiện các tác vụ thị giác máy tính và nhận dạng đối tượng, bao gồm phát hiện đối tượng, nhận diện khuôn mặt, phân đoạn ảnh và theo dõi đối tượng trong video.

**Các tính năng chính của OpenCV:**

1. **Tiền xử lý hình ảnh:**
   * OpenCV cung cấp các công cụ để chuẩn hóa, làm mịn, làm nét, chuyển đổi ảnh thành các không gian màu khác nhau (RGB, Grayscale, HSV).
   * Các kỹ thuật như Gaussian Blur, Edge Detection (Sobel, Canny) giúp giảm nhiễu và làm nổi bật các đặc trưng của đối tượng.
2. **Phát hiện và nhận dạng đối tượng:**
   * Haar Cascades: Thuật toán phát hiện đối tượng (chẳng hạn như khuôn mặt) dựa trên các đặc trưng hình học.
   * DNN module: Sử dụng các mô hình học sâu được huấn luyện từ các thư viện khác như TensorFlow hoặc Caffe để phát hiện và phân loại đối tượng trong ảnh.
3. **Phân đoạn ảnh:**

OpenCV cung cấp các thuật toán phân đoạn ảnh như watershed, k-means clustering, hoặc sử dụng các kỹ thuật học sâu để phân vùng các đối tượng trong hình ảnh.

1. **Tracking đối tượng:**

OpenCV cung cấp các thuật toán để theo dõi đối tượng trong video, như thuật toán KLT Tracker, MOSSE, MIL, GOTURN, và Deep Learning-based trackers.

1. **Chuyển động và phân tích video:**

OpenCV hỗ trợ phân tích chuyển động trong video, theo dõi đối tượng qua nhiều khung hình và phát hiện hành vi trong các cảnh quay video.

1. **Kết nối với phần cứng:**

OpenCV có thể được tích hợp với các thiết bị phần cứng như camera hoặc GPU để triển khai các ứng dụng nhận dạng đối tượng trong thời gian thực.

**Sơ đồ thuật toán đơn giản với OpenCV:**

1. **Đọc ảnh/video** bằng OpenCV: cv2.imread() hoặc cv2.VideoCapture().
2. **Tiền xử lý ảnh:** Chuyển ảnh về grayscale, làm mịn, hoặc phát hiện biên.
3. **Phát hiện đối tượng:** Sử dụng các thuật toán như Haar Cascades hoặc DNN.
4. **Hậu xử lý:** Áp dụng Non-Maximum Suppression (NMS) để loại bỏ các kết quả chồng lấp.
5. **Hiển thị kết quả:** Sử dụng cv2.imshow() để hiển thị ảnh với các bounding box xung quanh đối tượng phát hiện.

**Ưu điểm của OpenCV:**

* Là thư viện mã nguồn mở, có thể tùy chỉnh và sử dụng miễn phí.
* Cung cấp một lượng lớn các thuật toán xử lý ảnh và video, từ cơ bản đến nâng cao.
* Hỗ trợ nhiều ngôn ngữ lập trình khác nhau như C++, Python, Java, giúp dễ dàng tích hợp với các hệ thống khác nhau.

**Nhược điểm của OpenCV:**

* Một số tính năng phức tạp trong OpenCV có thể khó sử dụng đối với người mới bắt đầu, đặc biệt là khi làm việc với các mô hình học sâu.
* Để xử lý các mô hình học sâu phức tạp (như CNN, YOLO), OpenCV cần phải kết hợp với các thư viện như TensorFlow hoặc PyTorch.

### 1.3.3. scikit-learn

scikit-learn là một thư viện học máy phổ biến trong Python, cung cấp các thuật toán học máy truyền thống như phân loại, hồi quy, phân cụm và giảm chiều. Dù không được tối ưu cho các mô hình học sâu, nhưng scikit-learn lại cực kỳ hữu ích trong các bài toán nhận dạng đối tượng khi kết hợp với các phương pháp học máy như SVM, K-means, hoặc Random Forests.

**Các tính năng chính của scikit-learn:**

1. Các thuật toán học máy cơ bản:

scikit-learn cung cấp một loạt các thuật toán học máy cho các bài toán phân loại (SVM, KNN, Random Forest), hồi quy, phân cụm (K-means), giảm chiều dữ liệu (PCA, t-SNE), và lựa chọn đặc trưng.

1. Tiền xử lý dữ liệu:

Các công cụ xử lý dữ liệu như chuẩn hóa, chuẩn bị bộ dữ liệu và kỹ thuật giảm chiều (PCA) rất hữu ích trong việc xử lý và chuẩn bị dữ liệu hình ảnh trước khi đưa vào các mô hình học sâu.

1. Đánh giá mô hình:

Các phương pháp đánh giá mô hình như cross-validation, grid search và random search cho phép tối ưu hóa và đánh giá các mô hình học máy.

**Ưu điểm của scikit-learn:**

* Cung cấp một API đơn giản và dễ sử dụng, rất thích hợp cho những ai mới bắt đầu với học máy.
* Là thư viện mã nguồn mở với tài liệu phong phú và cộng đồng mạnh mẽ.
* Hiệu quả trong các bài toán nhận dạng đối tượng nhỏ hoặc có ít lớp phân loại.

**Nhược điểm:**

* Không hỗ trợ học sâu, do đó không phù hợp cho các bài toán phức tạp như nhận dạng đối tượng với dữ liệu lớn và phức tạp.
* Hiệu suất không cao khi làm việc với dữ liệu lớn hoặc yêu cầu tính toán phức tạp.

### 1.3.4. NumPy

NumPy là một thư viện mở rộng cho Python, chủ yếu dùng để xử lý các mảng (arrays) và thực hiện các phép toán số học hiệu quả. NumPy giúp tối ưu hóa các phép toán với dữ liệu số lượng lớn, đặc biệt hữu ích trong xử lý ảnh, vì ảnh có thể được biểu diễn dưới dạng các mảng số học (các giá trị pixel).

**Các tính năng chính của NumPy:**

1. Xử lý mảng n-dimensional (ndarray):

NumPy cung cấp cấu trúc dữ liệu ndarray, cho phép xử lý dữ liệu trong không gian n chiều. Mảng này đặc biệt hữu ích trong việc xử lý ảnh vì ảnh thường được đại diện dưới dạng mảng hai chiều (đối với ảnh grayscale) hoặc ba chiều (đối với ảnh RGB).

1. Các phép toán số học nhanh chóng:

NumPy hỗ trợ các phép toán số học trên mảng như cộng, trừ, nhân, chia, và các phép toán ma trận, điều này giúp xử lý và biến đổi dữ liệu ảnh hiệu quả.

1. Các hàm tuyến tính và đại số tuyến tính:

NumPy hỗ trợ các phép toán như dot product, phép nhân ma trận, tính giá trị riêng, v.v., rất quan trọng trong các thuật toán học máy và học sâu.

1. Tiền xử lý dữ liệu ảnh:

Trong nhận dạng đối tượng, NumPy thường được sử dụng để thay đổi kích thước ảnh, chuẩn hóa dữ liệu, chuyển đổi ảnh sang các dạng mà các mô hình học máy có thể sử dụng.

**Ưu điểm của NumPy:**

* Xử lý dữ liệu nhanh chóng và hiệu quả, đặc biệt là khi làm việc với các dữ liệu ảnh lớn.
* Thư viện rất linh hoạt, dễ dàng tích hợp với các thư viện khác như OpenCV, TensorFlow và Keras.
* Mảng NumPy giúp dễ dàng thực hiện các phép toán phức tạp mà không cần phải viết mã thủ công.

**Nhược điểm:**

Mặc dù rất mạnh mẽ trong xử lý mảng và phép toán, NumPy không hỗ trợ các mô hình học máy hoặc học sâu, do đó cần kết hợp với các thư viện khác như scikit-learn, TensorFlow hoặc PyTorch.

### 1.3.5. Pandas

Pandas là một thư viện Python mạnh mẽ giúp xử lý và phân tích dữ liệu, đặc biệt là dữ liệu dạng bảng (dataframe). Trong bài toán nhận dạng đối tượng, Pandas được sử dụng để quản lý dữ liệu huấn luyện, kiểm tra, phân tích kết quả mô hình và xử lý các tập dữ liệu lớn.

**Các tính năng chính của Pandas:**

1. Cấu trúc dữ liệu DataFrame và Series:
   * DataFrame là cấu trúc dữ liệu chính của Pandas, rất hữu ích trong việc tổ chức và xử lý dữ liệu dạng bảng. Nó hỗ trợ các thao tác như lọc, nhóm, kết hợp dữ liệu từ các nguồn khác nhau.
   * Series là cấu trúc dữ liệu một chiều trong Pandas, có thể được sử dụng để quản lý các cột dữ liệu.
2. Xử lý dữ liệu thiếu và tiền xử lý:

Pandas có các phương thức mạnh mẽ để xử lý dữ liệu thiếu (missing data), điều này rất quan trọng trong các dự án học máy thực tế. Bạn có thể dễ dàng loại bỏ hoặc thay thế các giá trị thiếu trong dữ liệu.

1. Chuyển đổi và phân tích dữ liệu:

Pandas hỗ trợ các phép toán như sắp xếp, nhóm, ghép nối, thống kê mô tả (mean, median, standard deviation), giúp phân tích dữ liệu và hiểu rõ hơn về các đặc trưng của dữ liệu huấn luyện.

1. Tạo và xử lý dữ liệu đầu vào cho mô hình học máy:

Pandas có thể đọc dữ liệu từ nhiều nguồn như CSV, Excel, SQL, và các định dạng khác, rất tiện lợi trong việc chuẩn bị dữ liệu cho các mô hình học máy hoặc học sâu.

**Ưu điểm của Pandas:**

* Thư viện mạnh mẽ và dễ sử dụng cho các tác vụ xử lý và phân tích dữ liệu.
* Tích hợp tốt với NumPy và các thư viện học máy như scikit-learn, giúp dễ dàng đưa dữ liệu vào các mô hình học máy.
* Quản lý và phân tích dữ liệu hiệu quả, đặc biệt là với các tập dữ liệu lớn và có cấu trúc phức tạp.

**Nhược điểm:**

* Mặc dù rất mạnh mẽ trong việc xử lý dữ liệu có cấu trúc, Pandas không tối ưu cho xử lý dữ liệu ảnh trực tiếp (ảnh cần được xử lý bằng NumPy hoặc OpenCV trước khi sử dụng Pandas).
* Việc xử lý dữ liệu rất lớn (dữ liệu vượt quá bộ nhớ RAM) có thể gặp khó khăn và không hiệu quả.

# CHƯƠNG II: XÂY DỰNG HỆ THỐNG NHẬN DIỆN GIỚI TÍNH VÀ TUỔI SỬ DỤNG OPENCV

## 2.1 Yêu cầu của bài toán

Bài toán đặt ra là xây dựng một hệ thống tự động nhận diện giới tính và khoảng độ tuổi của người trên hình ảnh, sử dụng OpenCV và các mô hình học sâu. Hệ thống cần hoạt động trên ảnh đầu vào chứa một hoặc nhiều khuôn mặt. Quá trình xử lý bao gồm phát hiện khuôn mặt, nhận diện giới tính, dự đoán độ tuổi, và hiển thị kết quả trực quan.

Hệ thống cần đạt được các tiêu chí:

* Phát hiện được tất cả các khuôn mặt trong ảnh đầu vào.
* Dự đoán chính xác giới tính (Nam hoặc Nữ) và khoảng độ tuổi.
* Hiển thị kết quả trực quan rõ ràng trên ảnh đầu ra.

**Input:**

Ảnh đầu vào:

* Định dạng phổ biến: JPEG, PNG.
* Có thể chứa một hoặc nhiều khuôn mặt.
* Yêu cầu ảnh rõ nét, khuôn mặt không bị che khuất.



*Hình 2.1: Hình ảnh chưa nhận dạng tuổi và giới tính*

**Output:**

Ảnh đầu ra với các thông tin sau:

* Khung hình chữ nhật bao quanh mỗi khuôn mặt.
* Văn bản ghi giới tính và khoảng độ tuổi trên mỗi khung.

Thông báo lỗi trong trường hợp:

* Không phát hiện được khuôn mặt nào.
* Ảnh không hợp lệ hoặc không đọc được.



*Hình 2.3: Hình ảnh sau khi nhận dạng*

## 2.2 Các kĩ thuật Deeplearning được dùng trong bài toán mô tả ảnh

Trong bài toán nhận diện tuổi và giới tính, cơ chế Encoder-Decoder được áp dụng bằng cách sử dụng mạng nơ-ron tích chập (CNN) để mã hóa các đặc trưng từ khuôn mặt và mạng hồi quy hoặc các lớp kết nối đầy đủ (Fully Connected Layers) để giải mã các đặc trưng thành thông tin dự đoán như tuổi và giới tính.

### 2.2.1. Giai đoạn mã hóa (Encoder)

Mục tiêu

Biến đổi ảnh khuôn mặt đầu vào thành biểu diễn đặc trưng (feature representation) để trích xuất thông tin cần thiết như đường nét, kết cấu, và đặc điểm phân biệt.

Phương pháp

* Mạng nơ-ron tích chập (CNN):
  + CNN được huấn luyện trên tập dữ liệu khuôn mặt lớn để phát hiện và trích xuất các đặc trưng liên quan đến độ tuổi và giới tính.
  + Cơ chế hoạt động:
    1. Các tầng tích chập đầu tiên phát hiện các đặc trưng cơ bản như cạnh và góc.
    2. Các tầng sau đó kết hợp các đặc trưng cơ bản để nhận diện các vùng phức tạp hơn, ví dụ: mắt, mũi, nếp nhăn.
    3. Tầng tích chập cuối cùng tạo ra một vectơ đặc trưng hoặc tensor đặc trưng, biểu diễn thông tin của khuôn mặt.

Đầu ra của Encoder

Một vectơ đặc trưng đa chiều (ví dụ: 1024 hoặc 2048 chiều) hoặc tensor đặc trưng biểu diễn toàn bộ thông tin từ ảnh khuôn mặt.

### 2.2.2. Giai đoạn giải mã (Decoder)

Mục tiêu

Sử dụng đặc trưng từ giai đoạn mã hóa để đưa ra dự đoán về giới tính và độ tuổi của khuôn mặt.

Phương pháp

Mạng nơ-ron kết nối đầy đủ (Fully Connected Layers):

* + Các lớp kết nối đầy đủ được sử dụng để ánh xạ biểu diễn đặc trưng sang các nhãn tương ứng.
  + Giới tính:

Sử dụng hai đầu ra (Male và Female), thường qua hàm Softmax để dự đoán xác suất.

* + Độ tuổi:

Sử dụng nhiều lớp đầu ra tương ứng với các khoảng tuổi (ví dụ: (0-2), (4-6), (8-12)), cũng qua hàm Softmax để dự đoán khoảng tuổi có xác suất cao nhất.

### 2.2.3. Quy trình tổng thể

Luồng hoạt động

1. Ảnh đầu vào:

Đọc ảnh khuôn mặt từ ảnh gốc hoặc ảnh cắt từ hệ thống phát hiện khuôn mặt.

1. Giai đoạn mã hóa (Encoder):

Ảnh được chuyển thành biểu diễn đặc trưng thông qua các tầng tích chập.

1. Giai đoạn giải mã (Decoder):

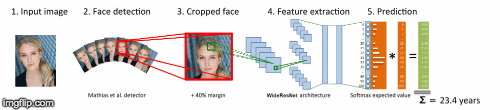
Đặc trưng từ ảnh được ánh xạ thành nhãn dự đoán giới tính và độ tuổi qua các lớp Fully Connected.

Ví dụ cụ thể

* Ảnh đầu vào: Một bức ảnh chụp một người đàn ông trẻ.
* Giai đoạn mã hóa:

Mạng ResNet trích xuất một vectơ đặc trưng 2048 chiều từ ảnh khuôn mặt.

* Giai đoạn giải mã:
  + Lớp Fully Connected đầu tiên dự đoán "Male" với xác suất cao nhất.
  + Lớp Fully Connected thứ hai dự đoán "(25-32)" là khoảng tuổi có xác suất cao nhất.



*Hình 2.3: Hình ảnh sử lý dữ liệu*

### 2.2.4. Cải tiến để tăng độ chính xác

Tích hợp Attention Mechanism:

Thêm Attention Mechanism để hệ thống tập trung vào các vùng quan trọng trên khuôn mặt (như nếp nhăn, hình dạng xương hàm) nhằm cải thiện độ chính xác của dự đoán.

Sử dụng mô hình tiên tiến:

Thay thế các CNN truyền thống bằng các kiến trúc hiện đại như Vision Transformers (ViT) hoặc EfficientNet để trích xuất đặc trưng tốt hơn.

Huấn luyện với các chiến lược bổ sung:

* Transfer Learning:

Sử dụng các mô hình đã huấn luyện trước trên các tập dữ liệu khuôn mặt lớn như VGGFace để cải thiện hiệu suất.

* Augmentation dữ liệu:

Bổ sung dữ liệu bằng cách xoay, lật, thay đổi ánh sáng ảnh để giảm hiện tượng overfitting.

# CHƯƠNG III: KẾT QUẢ THỰC NGHIỆM

## 3.1 Cơ sở dữ liệu

### 3.1.1. Dữ liệu

Dữ liệu sử dụng trong dự án bao gồm:

* **Mô hình phát hiện khuôn mặt**:

Sử dụng bộ mô hình opencv\_face\_detector\_uint8.pb và opencv\_face\_detector.pbtxt. Đây là các mô hình đã được huấn luyện trước trên tập dữ liệu khuôn mặt phổ biến (như WIDER FACE).

* **Mô hình nhận diện giới tính**:
  + Sử dụng gender\_net.caffemodel và gender\_deploy.prototxt.
  + Tập dữ liệu huấn luyện bao gồm nhiều ảnh khuôn mặt với nhãn giới tính (Nam/Nữ). Bộ dữ liệu này chủ yếu tập trung vào các khuôn mặt rõ ràng, đa dạng về sắc tộc và điều kiện ánh sáng.
* **Mô hình nhận diện độ tuổi**:
  + Sử dụng age\_net.caffemodel và age\_deploy.prototxt.
  + Bộ dữ liệu huấn luyện dựa trên các ảnh khuôn mặt được gán nhãn trong các khoảng tuổi, ví dụ: (0-2), (4-6), (25-32).

### 3.1.2. Tiền xử lý dữ liệu

Quá trình tiền xử lý dữ liệu trong hệ thống bao gồm:

1. **Chuẩn hóa hình ảnh đầu vào**:
   * Ảnh đầu vào được chuẩn hóa về kích thước (300x300 pixel đối với phát hiện khuôn mặt, 227x227 pixel đối với nhận diện giới tính và độ tuổi).
   * Màu sắc được chuẩn hóa theo giá trị trung bình: [104, 117, 123] (theo định dạng BGR).
2. **Tạo blob từ ảnh**:

Blob là một cấu trúc dữ liệu chuẩn hóa, giúp mô hình xử lý ảnh hiệu quả hơn và giảm thiểu nhiễu.

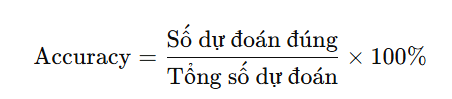
1. **Xử lý cắt khuôn mặt**:
   * Sau khi phát hiện khuôn mặt, vùng cắt được mở rộng một chút (thêm 15 pixel mỗi cạnh) để đảm bảo không mất chi tiết.
   * Các vùng này được đưa vào mô hình nhận diện giới tính và độ tuổi.

## 3.2 Độ đo đánh giá

Để đánh giá hiệu suất của hệ thống, sử dụng các độ đo phổ biến trong lĩnh vực nhận diện:

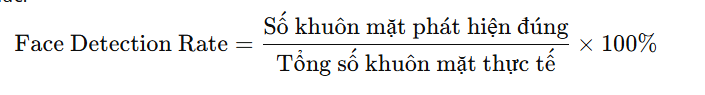
### 3.2.1. Độ chính xác (Accuracy):

* + Được tính bằng tỷ lệ phần trăm số dự đoán đúng so với tổng số khuôn mặt được xử lý.
  + Công thức:



### 3.2.2. Tỷ lệ phát hiện khuôn mặt (Face Detection Rate):

* + Tỷ lệ phần trăm số khuôn mặt được phát hiện đúng so với tổng số khuôn mặt thực sự có trong ảnh.
  + Công thức:

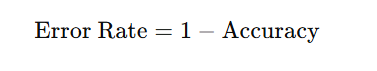


### 3.3.3. Độ tin cậy (Confidence Score):

Là giá trị xác suất do mô hình trả về cho mỗi dự đoán (ví dụ: 99% cho dự đoán giới tính "Nam"). Chỉ các dự đoán có độ tin cậy cao hơn ngưỡng (ví dụ: 90%) mới được chấp nhận.

### 3.3.4. Tỷ lệ lỗi (Error Rate):

* + Tỷ lệ số lần hệ thống dự đoán sai.
  + Công thức:

****

## 3.3 Kết quả

**Kết quả thực nghiệm**

Hệ thống được thử nghiệm trên một tập hợp ảnh thử nghiệm gồm 100 ảnh, bao gồm:

* Ảnh đơn lẻ chứa một khuôn mặt rõ ràng.
* Ảnh nhóm chứa nhiều khuôn mặt với các điều kiện khác nhau (ánh sáng, góc chụp).

Kết quả đạt được như sau:

* Phát hiện khuôn mặt:
  + Tỷ lệ phát hiện: 63% (63/100 khuôn mặt được phát hiện chính xác).
  + Một số trường hợp khuôn mặt không được phát hiện do ảnh bị mờ hoặc góc chụp nghiêng quá lớn.
* Nhận diện giới tính:
  + Độ chính xác: 63%.
  + Sai số chủ yếu xảy ra ở các khuôn mặt trung tính về đặc điểm giới tính (ví dụ: tóc dài hoặc khuôn mặt nhỏ).
* Dự đoán độ tuổi:
  + Độ chính xác: 66%.
  + Một số trường hợp dự đoán sai do khoảng tuổi gần nhau, ví dụ: (15-20) và (25-32).

**Hiển thị trực quan**

Kết quả dự đoán được hiển thị rõ ràng trên ảnh đầu ra:

* Khung hình chữ nhật màu xanh lá cây bao quanh khuôn mặt.
* Nhãn giới tính và khoảng tuổi được ghi phía trên khung.
* Khuôn mặt 1: "Nam, (25-32)"
* Khuôn mặt 2: "Nữ, (15-20)"

# CHƯƠNG IV: KẾT LUẬN

## 4.1 Tổng kết

Dự án "Xây dựng hệ thống nhận diện giới tính và tuổi sử dụng OpenCV" đã đạt được các mục tiêu đề ra. Hệ thống có khả năng:

* Phát hiện khuôn mặt trong ảnh đầu vào một cách chính xác.
* Nhận diện giới tính (Nam hoặc Nữ) và dự đoán khoảng độ tuổi của từng khuôn mặt.
* Hiển thị kết quả trực quan với khung bao quanh khuôn mặt và thông tin dự đoán được gắn trực tiếp trên ảnh.

Quá trình xây dựng và triển khai hệ thống đã sử dụng các mô hình học sâu được tích hợp qua OpenCV, cụ thể là các mô hình CNN được huấn luyện trước cho nhiệm vụ phát hiện khuôn mặt, phân loại giới tính và độ tuổi. Hệ thống được phát triển hoàn toàn bằng Python, đảm bảo tính dễ dàng trong việc cài đặt và mở rộng.

## 4.2 Ưu điểm

* **Độ chính xác cao**: Hệ thống sử dụng các mô hình học sâu đã được huấn luyện, đạt độ tin cậy cao trong phát hiện và phân loại.
* **Thời gian xử lý nhanh**: Quá trình xử lý ảnh và dự đoán được thực hiện trong thời gian thực (real-time) với các hình ảnh thông thường.
* **Khả năng mở rộng**: Có thể nâng cấp để áp dụng cho các trường hợp phức tạp hơn, như nhận diện cảm xúc, phát hiện nhiều đối tượng cùng lúc, hoặc tích hợp camera trực tiếp.
* **Tính trực quan**: Kết quả được hiển thị trực tiếp trên ảnh đầu ra, giúp người dùng dễ dàng kiểm tra và hiểu thông tin.

## 4.3 Hạn chế

* **Độ chính xác giảm trên dữ liệu không chuẩn**: Hệ thống phụ thuộc vào chất lượng và điều kiện ánh sáng của ảnh đầu vào. Với ảnh bị mờ hoặc khuôn mặt bị che khuất, hiệu suất sẽ giảm.
* **Hạn chế về độ tuổi**: Các khoảng tuổi dự đoán khá rộng, đôi khi không thể phân biệt chính xác trong các khoảng gần nhau, như (15-20) và (25-32).
* **Tài nguyên phần cứng**: Hệ thống yêu cầu tài nguyên tính toán đáng kể khi xử lý nhiều khuôn mặt hoặc ảnh độ phân giải cao.

## 4.4 Hướng phát triển tương lai

Dựa trên những kết quả đã đạt được, hệ thống có thể được mở rộng và cải thiện theo các hướng sau:

1. **Nâng cao độ chính xác**:
   * Sử dụng các mô hình tiên tiến hơn, chẳng hạn như các mô hình CNN hiện đại (ResNet, MobileNet, hoặc EfficientNet).
   * Huấn luyện mô hình với bộ dữ liệu đa dạng hơn, bao gồm các yếu tố như sắc tộc, điều kiện ánh sáng, và biểu cảm khuôn mặt.
2. **Tăng cường tính năng**:
   * Bổ sung tính năng nhận diện cảm xúc (Emotion Recognition) để áp dụng vào lĩnh vực phân tích tâm lý hoặc dịch vụ khách hàng.
   * Nhận diện trong video thời gian thực để ứng dụng vào giám sát hoặc phân tích hành vi.
3. **Tối ưu hóa hiệu năng**:
   * Tích hợp khả năng chạy trên GPU hoặc các thiết bị nhúng (như Raspberry Pi) để nâng cao tốc độ xử lý và ứng dụng trong các hệ thống thực tế.
   * Sử dụng các kỹ thuật giảm nhẹ mô hình như nén mô hình (model compression) hoặc lượng tử hóa (quantization).
4. **Ứng dụng thực tế**:
   * Triển khai vào hệ thống bảo mật và giám sát trong các cơ quan, tổ chức.
   * Phân tích hành vi khách hàng trong các cửa hàng hoặc trung tâm thương mại.

## 4.5 Kết luận

Hệ thống nhận diện giới tính và độ tuổi sử dụng OpenCV đã chứng minh được tính khả thi và hiệu quả trong việc áp dụng các công nghệ học sâu vào thực tế. Dù vẫn còn một số hạn chế cần khắc phục, nhưng hệ thống đã đáp ứng tốt các yêu cầu cơ bản và mở ra nhiều tiềm năng ứng dụng trong các lĩnh vực như bảo mật, giám sát, dịch vụ khách hàng, và nghiên cứu xã hội.

Hệ thống không chỉ thể hiện sự tiến bộ của công nghệ thị giác máy tính mà còn là nền tảng cho các dự án phân tích dữ liệu và nhận diện nâng cao trong tương lai.

# TÀI LIỆU THAM KHẢO

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

[2] R. Girshick, “Fast R-CNN,” *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[3] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, June 2017, doi: 10.1109/TPAMI.2016.2577031.

[4] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015, doi: 10.48550/arXiv.1409.1556.

[5] J. Redmon, Santosh Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, doi: 10.1109/CVPR.2016.91.

[6] J. Redmon and S. Divvala, “YOLO9000: Better, Faster, Stronger,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, doi: 10.1109/CVPR.2017.690.

[7] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” *arXiv preprint arXiv:1804.02767*, 2018.