**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 04: 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** | 20211344 | Ngô Ngọc Huy | DCCNTT12.10.5 |
| **2** | 20211228 | Hoàng Minh Hiếu | DCCNTT12.10.5 |
| **3** | 20211435 | Đào Quang Huy | DCCNTT12.10.5 |
| **4** | 20211359 | Chảo A Phúc | DCCNTT12.10.5 |
| **5** | 20211321 | Tô Duy Khánh | DCCNTT12.10.5 |

**Bắc Ninh, năm 2024**

**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 04: 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** | 20211344 | Ngô Ngọc Huy | DCCNTT12.10.5 |
| **2** | 20211228 | Hoàng Minh Hiếu | DCCNTT12.10.5 |
| **3** | 20211435 | Đào Quang Huy | DCCNTT12.10.5 |
| **4** | 20211359 | Chảo A Phúc | DCCNTT12.10.5 |
| **5** | 20211321 | Tô Duy Khánh | DCCNTT12.10.5 |

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

**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: 04**  **Tên học phần: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**  **Lớp Tín chỉ:**  **DCCNTT12.10.5** | |
| **Cán bộ chấm thi 1**  *(Ký và ghi rõ họ tên)*  Lương Thị Hồng Lan | **Cán bộ chấm thi 2**  *(Ký và ghi rõ họ tên)* |

| **TT** | **TIÊU CHÍ** | **THANG ĐIỂM** | **Ngô Ngọc Huy** | **Hoàng Minh Hiếu** | **Đào Quang Huy** | **Chảo A Phúc** | **Tô Duy Khánh** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **20211344** | **20211228** | **20211435** | **20211359** | **20211321** |
| **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

[**Chương 1: Tổng quan** 7](#_Toc184688155)

[**1.1. Bài toán nhận dạng:** 7](#_Toc184688156)

[1.1.1. Bài toán nhận dạng đối tượng: 7](#_Toc184688157)

[1.1.2.Một số ứng dụng: 8](#_Toc184688158)

[**1.2. Tổng quan các kỹ thuật áp dụng cho bài toán nhận dạng** 10](#_Toc184688159)

[1.2.1. Lớp mô hình họ CNN 10](#_Toc184688160)

[1.2.2. Lớp các mô hình họ YOLO 24](#_Toc184688161)

[**1.3. Ngôn ngữ lập trình** 28](#_Toc184688162)

[1.3.1.Python 28](#_Toc184688163)

[1.3.2.Thư viện sử dụng: 30](#_Toc184688164)

[**Chương 2: Xây dựng hệ thống nhận dạng** 33](#_Toc184688165)

[**2.1. Mô tả bài toán:** 33](#_Toc184688166)

[2.1.1 Input 33](#_Toc184688167)

[2.1.2. Phương pháp áp dụng: 33](#_Toc184688168)

[**2.2. Xây dựng hệ thống** 35](#_Toc184688169)

[**Chương 3: Kết quả thực nghiệm** 39](#_Toc184688170)

[**3.1 Cơ sở dữ liệu** 39](#_Toc184688171)

[3.1.1. Dữ liệu 39](#_Toc184688172)

[3.1.2. Tiền xử lý dữ liệu 39](#_Toc184688173)

[**3.2 Độ đo đánh giá** 40](#_Toc184688174)

[3.2.1. Độ chính xác (Accuracy): 40](#_Toc184688175)

[3.2.2. Tỷ lệ phát hiện khuôn mặt (Face Detection Rate): 40](#_Toc184688176)

[3.3.3. Độ tin cậy (Confidence Score): 41](#_Toc184688177)

[3.3.4. Tỷ lệ lỗi (Error Rate): 41](#_Toc184688178)

[**3.3 Kết quả** 41](#_Toc184688179)

[**Tài liệu tham khảo** 44](#_Toc184688180)

# **Chương 1: Tổng quan**

## **1.1. Bài toán nhận dạng:**

### 1.1.1. Bài toán nhận dạng đối tượng:

Nhận dạng đối tượng là một thuật ngữ chung để mô tả một tập hợp các nhiệm vụ thị giác máy tính có liên quan liên quan đến việc xác định các đối tượng trong ảnh kỹ thuật số.

Phân loại hình ảnh liên quan đến việc dự đoán lớp của một đối tượng trong một hình ảnh. Định vị vật thể đề cập đến việc xác định vị trí của một hoặc nhiều đối tượng trong một hình ảnh và vẽ bounding box xung quanh chúng. Phát hiện đối tượng kết hợp hai nhiệm vụ trên và thực hiện cho một hoặc nhiều đối tượng trong hình ảnh. Chúng ta có thể phân biệt giữa ba nhiệm vụ thị giác máy tính cơ bản trên thông qua input và output của chúng như sau:

- Phân loại hình ảnh: Dự đoán nhãn của một đối tượng trong một hình ảnh.

+ Input: Một hình ảnh với một đối tượng, chẳng hạn như một bức ảnh.

+ Output: Nhãn lớp (ví dụ: một hoặc nhiều số nguyên được ánh xạ tới nhãn lớp).

- Định vị đối tượng: Xác định vị trí hiện diện của các đối tượng trong ảnh và cho biết vị trí của chúng bằng bounding box.

+ Input: Một hình ảnh có một hoặc nhiều đối tượng, chẳng hạn như một bức ảnh.

+ Output: Một hoặc nhiều bounding box được xác định bởi tọa độ tâm, chiều rộng và chiều cao.

- Phát hiện đối tượng: Xác định vị trí hiện diện của các đối tượng trong bounding box và nhãn của các đối tượng nằm trong một hình ảnh.

+ Input: Một hình ảnh có một hoặc nhiều đối tượng, chẳng hạn như một bức ảnh.

+ Output: Một hoặc nhiều bounding box và nhãn cho mỗi bounding box.

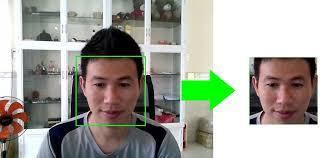
Một số định nghĩa khác cũng rất quan trọng trong computer vision là phân đoạn đối tượng (object segmentation), trong đó các đối tượng được nhận dạng bằng cách làm nổi bật các pixel cụ thể của đối tượng thay vì bounding box. Và image captioning kết hợp giữa các kiến trúc mạng CNN và LSTM để đưa ra các lý giải về hành động hoặc nội dung của một bức ảnh.

### 1.1.2.Một số ứng dụng:

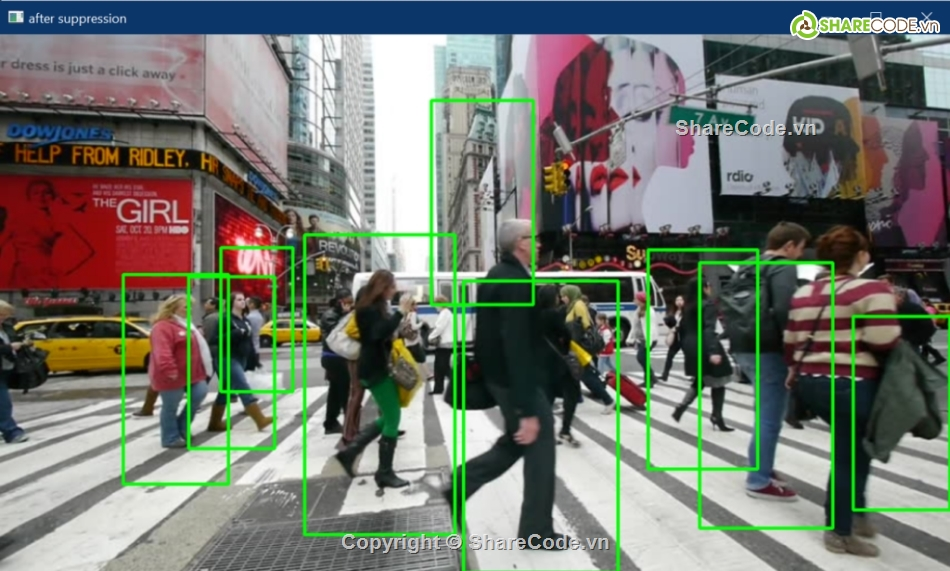
- Nhận dạng biển số – sử dụng cả công nghệ phát hiện đối tượng và nhận dạng ký tự quang học (OCR) để nhận dạng các ký tự chữ và số trên biển số xe. Tính năng phát hiện đối tượng được sử dụng thông qua việc lưu giữ lại hình ảnh và phát hiện các đối tượng cụ thể như xe cộ, phương tiện đi lại trên bức ảnh đó. Sau khi đã xác định được tọa độ của phương tiện trên ảnh, chúng ta có thể sử dụng các mô hình Object Detection khác để phát hiện biển số xe và các ký tự trên biển số. Từ đó, áp dụng các mô hình OCR hoặc image classification để nhận dạng ký tự trên hình, ánh xạ từ dạng dữ liệu hình ảnh sang dạng chữ (text).



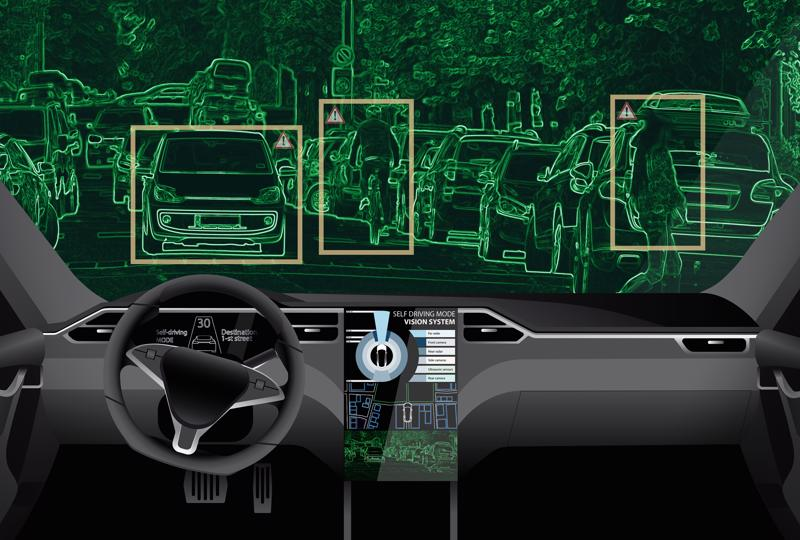
- Phát hiện và nhận dạng khuôn mặt – một trong những ứng dụng chính của phát hiện đối tượng là nhận diện và phát hiện khuôn mặt. Với sự trợ giúp của các thuật toán hiện đại, chúng ta có thể phát hiện khuôn mặt người trong một hình ảnh hoặc video.



- Theo dõi đối tượng – Công nghệ này có thể ứng dụng để theo dõi chuyển động của một đối tượng hay đồ vật cụ thể. Chẳng hạn, trong khi xem một trận bóng chày hoặc cricket, quả bóng có thể bị đập ra xa. Trong những tình huống này, thuật toán giúp chúng ta phát hiện, nhận dạng và theo dõi (tracking) vị trí và đường bay của quả bóng hiện tại.



- Ô tô tự lái – đối với ô tô tự lái, điều quan trọng là phải nghiên cứu các yếu tố môi trường xung quanh ô tô khi lái xe. Một mô hình phát hiện đối tượng được đào tạo trên nhiều đối tượng để phát hiện các thực thể khác nhau cản trở việc lái xe nhằm đảo bảo an toàn cho người lái.



- Người máy – Hiện nay thuật toán phát hiện đối tượng thường được ứng dụng trong robot để nhận diện và di chuyển các vật nặng, giảm thiểu những công việc về thể chất cho con người và tự động hóa công việc.



## **1.2. Tổng quan các kỹ thuật áp dụng cho bài toán nhận dạng**

### 1.2.1. Lớp mô hình họ CNN

1.2.1.1. CNN

Mạng CNN là một tập hợp các lớp Convolution chồng lên nhau và sử dụng các hàm nonlinear activation như ReLU và tanh để kích hoạt các trọng số trong các node. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo.

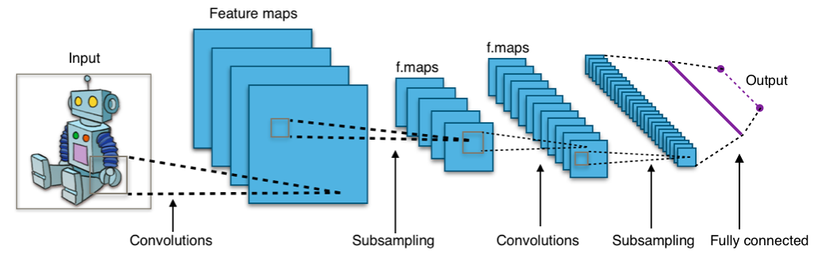
Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo. Trong mô hình mạng truyền ngược (feedforward neural network) thì mỗi neural đầu vào (input node) cho mỗi neural đầu ra trong các lớp tiếp theo.

Mô hình này gọi là mạng kết nối đầy đủ (fully connected layer) hay mạng toàn vẹn (affine layer). Còn trong mô hình CNNs thì ngược lại. Các layer liên kết được với nhau thông qua cơ chế convolution.

Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Như vậy mỗi neuron ở lớp kế tiếp sinh ra từ kết quả của filter áp đặt lên một vùng ảnh cục bộ của neuron trước đó.

Mỗi một lớp được sử dụng các filter khác nhau thông thường có hàng trăm hàng nghìn filter như vậy và kết hợp kết quả của chúng lại. Ngoài ra có một số layer khác như pooling/subsampling layer dùng để chắt lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu).

Trong quá trình huấn luyện mạng (traning) CNN tự động học các giá trị qua các lớp filter dựa vào cách thức mà bạn thực hiện. Ví dụ trong tác vụ phân lớp ảnh, CNNs sẽ cố gắng tìm ra thông số tối ưu cho các filter tương ứng theo thứ tự raw pixel > edges > shapes > facial > high-level features. Layer cuối cùng được dùng để phân lớp ảnh.



*Hình 1.2.1.1: Hình ảnh mô tả kỹ thuật CNN*

Trong mô hình CNN có 2 khía cạnh cần quan tâm là tính bất biến (Location Invariance) và tính kết hợp (Compositionality). Với cùng một đối tượng, nếu đối tượng này được chiếu theo các gốc độ khác nhau (translation, rotation, scaling) thì độ chính xác của thuật toán sẽ bị ảnh hưởng đáng kể.

Pooling layer sẽ cho bạn tính bất biến đối với phép dịch chuyển (translation), phép quay (rotation) và phép co giãn (scaling). Tính kết hợp cục bộ cho ta các cấp độ biểu diễn thông tin từ mức độ thấp đến mức độ cao và trừu tượng hơn thông qua convolution từ các filter.

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| Tính bất biến (Location Invariance):  CNN có khả năng xử lý tốt các biến đổi như dịch chuyển (translation), xoay (rotation), và co giãn (scaling) nhờ sử dụng các cơ chế như pooling layer, giúp giảm sự nhạy cảm đối với các thay đổi nhỏ trong đầu vào. | Yêu cầu tài nguyên tính toán lớn:  Để huấn luyện các mạng CNN lớn với nhiều lớp, cần sử dụng GPU hoặc các phần cứng chuyên dụng. Điều này có thể tốn kém và yêu cầu cao về hạ tầng. |
| Tính kết hợp (Compositionality):  Các lớp convolution tạo ra cấp độ biểu diễn từ thấp (raw pixel) đến cao (high-level features), giúp mạng học được các thông tin trừu tượng hơn theo thời gian. | Nhạy cảm với lượng dữ liệu lớn:  CNN cần một lượng dữ liệu đủ lớn để học được các mẫu đặc trưng. Với các tập dữ liệu nhỏ, mạng dễ gặp hiện tượng overfitting. |
| Khả năng trích xuất đặc trưng tự động:  Không cần phải thiết kế thủ công các đặc trưng (features) như trong các thuật toán truyền thống, CNN tự động học các đặc trưng phù hợp với bài toán. | Không tối ưu với các bài toán phi không gian:  Mạng CNN không phù hợp với các loại dữ liệu không có mối quan hệ không gian rõ ràng, như dữ liệu dạng chuỗi (sequence) hay dữ liệu bảng (tabular data). |
| Khả năng mở rộng:  Mạng có thể được mở rộng với các lớp sâu hơn để giải quyết các bài toán phức tạp hơn như phân loại ảnh, nhận dạng khuôn mặt, hoặc phát hiện đối tượng. | Khó giải thích (Interpretability):  Mặc dù CNN hoạt động hiệu quả, việc hiểu rõ lý do mạng đưa ra một quyết định cụ thể là rất khó, làm giảm tính minh bạch trong một số ứng dụng nhạy cảm. |
| Giảm số lượng tham số:  Nhờ cơ chế kết nối cục bộ (local connectivity) và chia sẻ trọng số (weight sharing), CNN giảm đáng kể số lượng tham số cần học so với mạng truyền thống (Fully Connected Networks), giúp cải thiện hiệu quả tính toán và giảm nguy cơ overfitting. | Tính bất biến còn giới hạn:  Dù pooling layer giúp tăng tính bất biến đối với các phép dịch chuyển hoặc co giãn, nhưng các biến đổi lớn (như xoay góc lớn hoặc thay đổi tỷ lệ lớn) vẫn có thể làm giảm hiệu quả của mạng. |
| Hiệu quả với dữ liệu không gian (Spatial Data):  CNN đặc biệt phù hợp với các bài toán liên quan đến dữ liệu không gian, như hình ảnh hoặc video, nhờ vào cấu trúc lớp convolution. | Phụ thuộc vào việc thiết kế kiến trúc:  Hiệu quả của CNN phụ thuộc rất nhiều vào thiết kế kiến trúc mạng (số lớp, kích thước filter, loại pooling, v.v.). Điều này đòi hỏi kinh nghiệm và kiến thức chuyên môn từ người phát triển. |

Bảng 1.2.1.1: Ưu nhược điểm của CNN

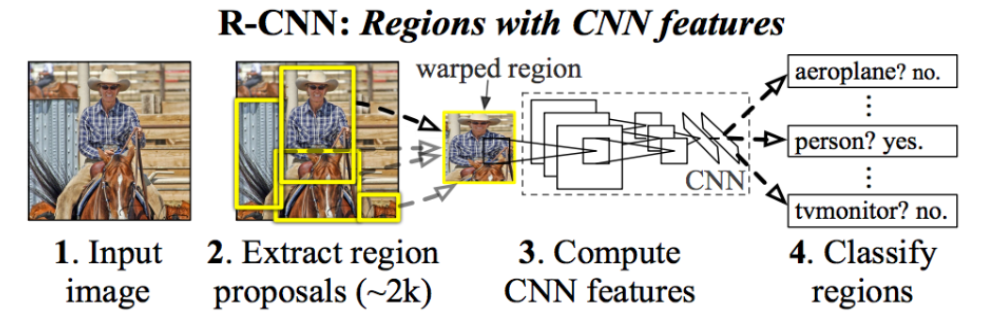
1.2.1.2. R-CNN (2014)

Nó có thể là một trong những ứng dụng nền móng đầu tiên của mạng nơ ron tích chập đối với vấn đề định vị, phát hiện và phân đoạn đối tượng. Cách tiếp cận đã được chứng minh trên các bộ dữ liệu điểm chuẩn, đạt được kết quả tốt nhất trên bộ dữ liệu VOC-2012 và bộ dữ liệu phát hiện đối tượng ILSVRC-2013 gồm 200 lớp [1].

Kiến trúc của R-CNN gồm 3 thành phần đó là:

* Vùng đề xuất hình ảnh (Region proposal): Có tác dụng tạo và trích xuất các vùng đề xuất chứa vật thể được bao bởi các bounding box [1].
* Trích lọc đặc trưng (Feature Extraction): Trích xuất các đặc trưng giúp nhận diện hình ảnh từ các region proposal thông qua các mạng deep convolutional neural network [1].
* Phân loại (classifier): Dựa vào input là các features ở phần trước để phân loại hình ảnh chứa trong region proposal về đúng nhãn [1].

Kiến trúc của mô hình được mô tả trong biểu đồ bên dưới:



*Hình 1.2.1.2: Hình ảnh mô tả kỹ thuật R-CNN*

Sơ đồ pipeline xử lý trong mô hình mạng R-CNN (được trích xuất từ bài báo gốc). Ta có thể nhận thấy các hình ảnh con được trích xuất tại bước 2 với số lượng rất lớn (khoảng 2000 region proposals). Tiếp theo đó áp dụng một mạng deep CNN để tính toán các feature tại bước 3 và trả ra kết quả dự báo nhãn ở bước 4 như một tác vụ image classification thông thường [1].

Một kỹ thuật được sử dụng để đề xuất các region proposal hoặc các bounding box chứa các đối tượng tiềm năng trong hình ảnh được gọi là “selective search”, các region proposal có thể được phát hiện bởi đa dạng những thuật toán khác nhau. Nhưng điểm chung là đều dựa trên tỷ lệ IoU giữa bounding box và ground truth box mà bạn đọc sẽ được tìm hiểu ở bài viết tiếp theo giới thiệu về mạng SSD.

Trích xuất đặc trưng về bản chất là một mạng CNN học sâu, ở đây là AlexNet, mạng đã giành chiến thắng trong cuộc thi phân loại hình ảnh ILSVRC-2012. Đầu ra của CNN là một vectơ 4096 chiều mô tả nội dung của hình ảnh được đưa đến một mô hình SVM tuyến tính để phân loại.

Đây là một ứng dụng tương đối đơn giản và dễ hiểu của CNN đối với vấn đề object localization và object detection. Một nhược điểm của phương pháp này là chậm, đòi hỏi phải vượt qua nhiều module độc lập trong đó có trích xuất đặc trưng từ một mạng CNN học sâu trên từng region proposal được tạo bởi thuật toán đề xuất vùng chứa ảnh. Đây là một vấn đề chính cần giải quyết vì bài viết mô tả mô hình hoạt động trên khoảng 2000 vùng được đề xuất cho mỗi hình ảnh tại thời điểm thử nghiệm [1].

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| Phát hiện đối tượng chính xác : R-CNN cung cấp khả năng phát hiện đối tượng chính xác bằng cách tận dụng các tính năng tích chập dựa trên vùng. Nó vượt trội trong các tình huống mà việc định vị và nhận dạng đối tượng chính xác là rất quan trọng. | Độ phức tạp tính toán : R-CNN có khối lượng tính toán lớn. Nó bao gồm việc trích xuất các đề xuất vùng, áp dụng CNN cho từng đề xuất, sau đó chạy các tính năng đã trích xuất thông qua một bộ phân loại. Quá trình nhiều giai đoạn này có thể chậm và đòi hỏi nhiều tài nguyên. |
| Độ bền trước các biến thể đối tượng : Mô hình R-CNN có thể xử lý các đối tượng có nhiều kích thước, hướng và tỷ lệ khác nhau, phù hợp với các tình huống thực tế có nhiều đối tượng và bối cảnh phức tạp. | Suy luận chậm : Do xử lý tuần tự các đề xuất vùng, R-CNN tương đối chậm trong quá trình suy luận. Các ứng dụng thời gian thực có thể thấy độ trễ này không thể chấp nhận được. |
| Tính linh hoạt : R-CNN là một khuôn khổ linh hoạt có thể được điều chỉnh cho nhiều tác vụ phát hiện đối tượng khác nhau, bao gồm phân đoạn thể hiện và theo dõi đối tượng. Bằng cách sửa đổi các lớp cuối cùng của mạng, bạn có thể tùy chỉnh R-CNN để phù hợp với nhu cầu cụ thể của mình. | Đề xuất vùng chồng lấn : R-CNN có thể tạo ra nhiều đề xuất vùng chồng lấn đáng kể, dẫn đến tính toán dư thừa và có khả năng ảnh hưởng đến hiệu suất phát hiện. |

Bảng 1.2.1.2: Ưu nhược điểm của R-CNN (2014)

*1.2.1.3. Fast R-CNN (2015)*

Dựa trên thành công của R-CNN, Ross Girshick (lúc này đã chuyển sang Microsoft Research) đề xuất một mở rộng để giải quyết vấn đề của R-CNN trong một bài báo vào năm 2015 với tiêu đề rất ngắn gọn [Fast R-CNN](https://arxiv.org/abs/1504.08083).

Bài báo chỉ ra những hạn chế của R-CNN đó là:

* Training qua một pipeline gồm nhiều bước: Pipeline liên quan đến việc chuẩn bị và vận hành ba mô hình riêng biệt [2].
* Chi phí training tốn kém về số lượng bounding box và thời gian huấn luyện: Mô hình huấn luyện một mạng CNN học sâu trên rất nhiều region proposal cho mỗi hình ảnh nên rất chậm [2].
* Phát hiện đối tượng chậm: Tốc độ xử lý không thể đảm bảo realtime [2].

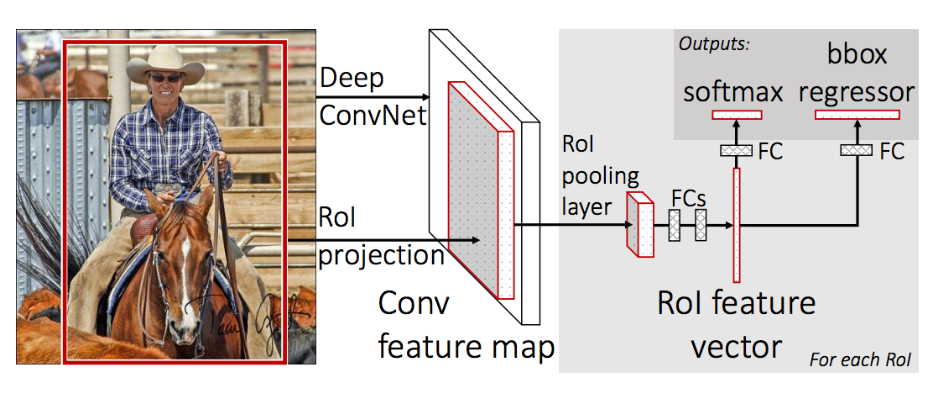
Trước đó một bài báo đã đề xuất phương pháp để tăng tốc kỹ thuật được gọi là [mạng tổng hợp kim tự tháp - Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition](https://arxiv.org/abs/1406.4729), hoặc SPPnets vào năm 2014. Phương pháp này đã tăng tốc độ trích xuất features nhờ lan truyền thuận trên bộ nhớ đệm.

Điểm đột phá của Fast R-CNN là sử dụng một single model thay vì pipeline để phát hiện region và classification cùng lúc [2].

Kiến trúc của mô hình trích xuất từ bức ảnh một tập hợp các region proposals làm đầu vào được truyền qua mạng deep CNN. Một pretrained-CNN, chẳng hạn VGG-16, được sử dụng để trích lọc features. Phần cuối của deep-CNN là một custom layer được gọi là layer vùng quan tâm (Region of Interest Pooling - RoI Pooling) có tác dụng trích xuất các features cho một vùng ảnh input nhất định [2].

Sau đó các features được kết bởi một lớp fully connected. Cuối cùng mô hình chia thành hai đầu ra, một đầu ra cho dự đoán nhãn thông qua một softmax layer và một đầu ra khác dự đoán bounding box (kí hiệu là bbox) dựa trên hồi qui tuyến tính. Quá trình này sau đó được lặp lại nhiều lần cho mỗi vùng RoI trong một hình ảnh [2].

Kiến trúc của mô hình được tóm tắt trong hình dưới đây, được lấy từ bài báo.



*Hình 1.2.1.3: Hình ảnh minh họa kỹ thuật Fast R-CNN (2015)*

Kiến trúc single model Fast R-CNN (được trích xuất từ bài báo gốc). Ở bước đầu ta áp dụng một mạng Deep CNN để trích xuất ra feature map. Thay vì warp image của region proposal như ở R-CNN chúng ta xác dịnh ngay vị trí hình chiếu của của region proposal trên feature map thông qua phép chiếu RoI projection. Vị trí này sẽ tương đối với vị trí trên ảnh gốc. Sau đó tiếp tục truyền output qua các layer RoI pooling layer và các Fully Connected layers để thu được RoI feature véc tơ. Sau đó kết quả đầu ra sẽ được chia làm 2 nhánh. 1 Nhánh giúp xác định phân phối xác suất theo các class của 1 vùng quan tâm RoI thông qua hàm softmax và nhánh còn xác định tọa độ của bounding box thông qua hồi qui các offsets [2].

Mô hình này nhanh hơn đáng kể cả về huấn luyện và dự đoán, tuy nhiên vẫn cần một tập hợp các region proposal được đề xuất cùng với mỗi hình ảnh đầu vào.

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| **Tốc độ nhanh hơn:** Fast R-CNN cải thiện đáng kể tốc độ bằng cách xử lý một lần ảnh đầu vào và trích xuất đặc trưng cho tất cả các vùng đề xuất. | **Yêu cầu chọn vùng:** Fast R-CNN vẫn phụ thuộc vào việc chọn vùng đề xuất (region proposal) để hoạt động, thường sử dụng Selective Search, có thể không tối ưu. |
| **Chất lượng chính xác cao:** Độ chính xác được cải thiện nhờ việc kết hợp toàn bộ ảnh vào quá trình học và phân loại vùng. | **Không hoàn toàn tự động:** Vẫn cần một bước ngoại vi để tạo ra các vùng đề xuất, không thể tự động như Faster R-CNN. |
| **Hỗ trợ học một giai đoạn:** Cho phép việc học một giai đoạn, giúp giảm số lượng tham số và thời gian huấn luyện. | **Khó khăn trong việc xử lý ảnh lớn:** Với ảnh kích thước lớn, Fast R-CNN có thể gặp khó khăn trong việc xử lý do giới hạn về kích thước đầu vào. |
| **Giảm thiểu bộ nhớ:** Sử dụng các kỹ thuật tối ưu như ROI Pooling để giảm lượng bộ nhớ cần thiết cho việc lưu trữ thông tin vùng. | **Cần điều chỉnh kích thước:** Kích thước của vùng đề xuất cần được điều chỉnh để phù hợp với quy trình phân loại, có thể làm giảm tính linh hoạt. |

Bảng 1.2.1.4: Ưu nhược điểm của Fast R-CNN (2015)

1.2.1.3. Faster R-CNN (2016)

Kiến trúc mô hình đã được cải thiện hơn nữa về cả tốc độ huấn luyện và phát hiện được đề xuất bởi Shaoqing Ren và các cộng sự tại Microsoft Research trong bài báo năm 2016 có tiêu đề [Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks](https://arxiv.org/abs/1506.01497). Dịch nghĩa là “Faster R-CNN: Hướng tới phát hiện đối tượng theo thời gian thực với các mạng đề xuất khu vực”.

Kiến trúc này mang lại độ chính xác cao nhất đạt được trên cả hai nhiệm vụ phát hiện và nhận dạng đối tượng tại các cuộc thi ILSVRC-2015 và MS COCO-2015.

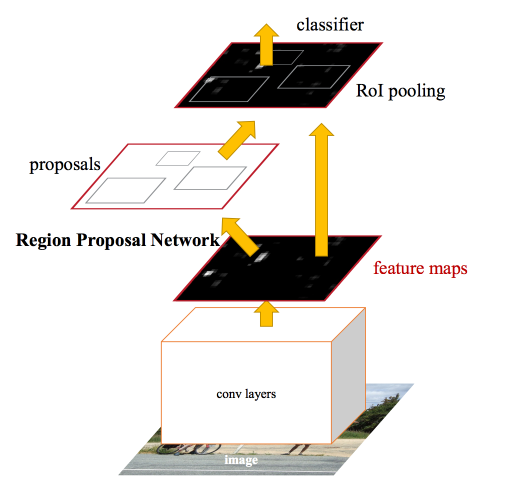
Kiến trúc được thiết kế để đề xuất và tinh chỉnh các region proposals như là một phần của quá trình huấn luyện, được gọi là mạng đề xuất khu vực (Region Proposal Network), hoặc RPN. Các vùng này sau đó được sử dụng cùng với mô hình Fast R-CNN trong một thiết kế mô hình duy nhất. Những cải tiến này vừa làm giảm số lượng region proposal vừa tăng tốc hoạt động trong thời gian thử nghiệm mô hình lên gần thời gian thực với hiệu suất tốt nhất. Tốc độ là 5fps trên một GPU [3].

Mặc dù là một mô hình đơn lẻ duy nhất, kiến trúc này là kết hợp của hai modules:

* Mạng đề xuất khu vực (Region Proposal Network, viết tắT là RPN). Mạng CNN để đề xuất các vùng và loại đối tượng cần xem xét trong vùng [3].
* Fast R-CNN: Mạng CNN để trích xuất các features từ các region proposal và trả ra các bounding box và nhãn [3].

Cả hai modules hoạt động trên cùng một output của một mạng deep CNN. Mạng RPN hoạt động như một cơ chế attention cho mạng Fast R-CNN, thông báo cho mạng thứ hai về nơi cần xem hoặc chú ý.

Kiến trúc của mô hình được tổng kết thông qua sơ đồ bên dưới:



*Hình 1.2.1.4: Hình ảnh minh họa kỹ thuật Faster R-CNN (2016)*

Kiến trúc mô hình Faster R-CNN (được trích xuất từ bài báo gốc). Ở giai đoạn sớm sử dụng một mạng deep CNN để tạo ra một feature map. Khác với Fast R-CNN, kiến trúc này không tạo RoI ngay trên feature map mà sử dụng feature map làm đầu vào để xác định các region proposal thông qua một RPN network. Đồng thời feature maps cũng là đầu vào cho classifier nhằm phân loại các vật thể của region proposal xác định được từ RPN network [3].

RPN hoạt động bằng cách lấy đầu ra của một mạng pretrained deep CNN, chẳng hạn như VGG-16, và truyền feature map vào một mạng nhỏ và đưa ra nhiều region proposals và nhãn dự đoán cho chúng. Region proposals là các bounding boxes, dựa trên các anchor boxes hoặc hình dạng được xác định trước được thiết kế để tăng tốc và cải thiện khả năng đề xuất vùng. Dự đoán của nhãn được thể hiện dưới dạng nhị phân cho biết region proposal có xuất hiện vật thể hoặc không [3].

Một quy trình huấn luyện xen kẽ được sử dụng trong đó cả hai mạng con được đào tạo cùng một lúc. Điều này cho phép các tham số trong feature dectector của deep CNN được tinh chỉnh cho cả hai tác vụ cùng một lúc [3].

Tại thời điểm viết, kiến trúc Faster R-CNN này là đỉnh cao của họ model R-CNN và tiếp tục đạt được kết quả gần như tốt nhất trong các nhiệm vụ nhận diện đối tượng. Một mô hình mở rộng hỗ trợ cho phân đoạn hình ảnh, được mô tả trong bài báo năm 2017 có tựa đề [Mask R-CNN](https://arxiv.org/abs/1703.06870).

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| Hiệu suất cao:Faster R-CNN đạt độ chính xác vượt trội trong phát hiện và nhận dạng đối tượng nhờ khả năng kết hợp Region Proposal Network (RPN) và Fast R-CNN vào cùng một kiến trúc. | Tốc độ chưa đạt mức thời gian thực hoàn toàn: Mặc dù nhanh hơn các mô hình trước, tốc độ 5 fps trên GPU vẫn chưa đủ để đáp ứng các ứng dụng đòi hỏi thời gian thực như xe tự lái hay camera giám sát. |
| Tích hợp Region Proposal Network (RPN)**:** RPN giúp giảm đáng kể số lượng các region proposals không liên quan, đồng thời tập trung vào các khu vực có khả năng chứa đối tượng.  Khả năng đề xuất vùng theo thời gian thực nhờ việc tối ưu hóa qua các anchor boxes, giảm độ phức tạp và thời gian xử lý. | Yêu cầu phần cứng mạnh: Faster R-CNN cần GPU hiệu suất cao để đạt tốc độ tối ưu, điều này có thể là rào cản trong các môi trường tính toán hạn chế. |
| Thiết kế kiến trúc liền mạch:Cả RPN và Fast R-CNN đều sử dụng chung feature map, tiết kiệm tài nguyên tính toán so với các phương pháp trước đó như R-CNN và Fast R-CNN. Mô hình duy nhất nhưng xử lý được cả hai nhiệm vụ: xác định vùng và phân loại đối tượng. | Độ phức tạp trong thiết kế và huấn luyện:Quy trình huấn luyện xen kẽ (RPN và Fast R-CNN) đòi hỏi sự đồng bộ và tối ưu hóa cẩn thận, khiến việc triển khai phức tạp hơn so với các mô hình đơn giản hơn như YOLO hoặc SSD. |
| Khả năng mở rộng:Faster R-CNN có thể được mở rộng để hỗ trợ các nhiệm vụ như phân đoạn đối tượng (Mask R-CNN), giúp mở rộng ứng dụng vào các lĩnh vực như y tế, giao thông, và robot. | Không phù hợp với các ứng dụng yêu cầu tốc độ rất cao: Các mô hình như YOLO (You Only Look Once) hoặc SSD (Single Shot MultiBox Detector) thường được chọn cho các ứng dụng cần tốc độ cực nhanh (lên đến hàng chục fps) dù độ chính xác có thể thấp hơn. |
| Xử lý thời gian thực (tương đối):Tốc độ cải thiện rõ rệt, đạt 5 fps trên GPU, nhanh hơn so với các phương pháp R-CNN và Fast R-CNN ban đầu. | Phụ thuộc vào anchor boxes: Cơ chế anchor boxes cần phải được thiết kế tốt và phù hợp với từng bộ dữ liệu cụ thể. Điều này làm tăng khối lượng công việc để điều chỉnh mô hình cho các ứng dụng khác nhau. |

Bảng 1.2.1.4: Ưu nhược điểm Faster R-CNN (2016)

### 1.2.2. Lớp các mô hình họ YOLO

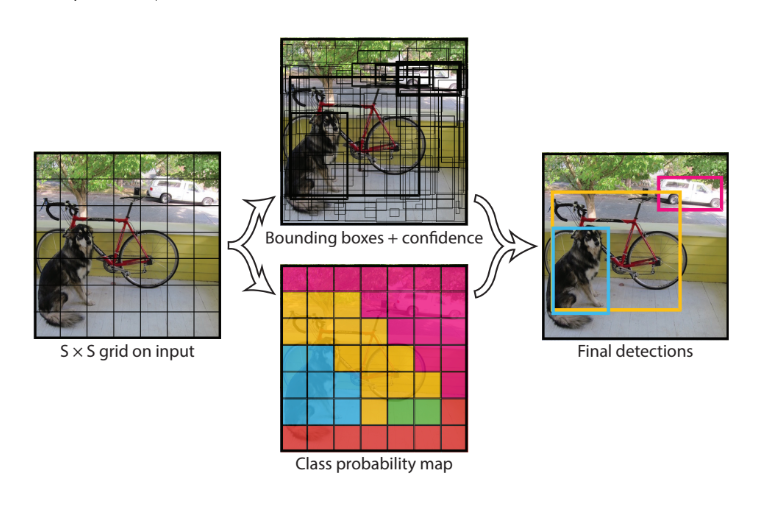
1.2.2.1. YOLO (2015)

Mô hình YOLO được mô tả lần đầu tiên bởi Joseph Redmon, và các cộng sự. trong bài viết năm 2015 có tiêu đề [Bạn chỉ nhìn một lần: Phát hiện đối tượng theo thời gian thực - You Only Look Once: Unified, Real-Time Object Detection](https://arxiv.org/abs/1506.02640). Trong công trình này thì một lần nữa Ross Girshick, người phát triển mạng R-CNN, cũng là một tác giả và người đóng góp khi ông chuyển qua [Facebook AI Research](https://research.fb.com/category/facebook-ai-research/).

Phương pháp chính dựa trên một mạng neural network duy nhất được huấn luyện dạng end-to-end model. Mô hình lấy input là một bức ảnh và dự đoán các bounding box và nhãn lớp cho mỗi bounding box. Do không sử dụng region proposal nên kỹ thuật này có độ chính xác thấp hơn (ví dụ: nhiều lỗi định vị vật thể - localization error hơn), mặc dù hoạt động ở tốc độ 45 fps (khung hình / giây) và tối đa 155 fps cho phiên bản tối ưu hóa tốc độ. Tốc độ này còn nhanh hơn cả tốc độ khung hình của máy quay phim thông thường chỉ vào khoảng 24 fps [5].

Mô hình hoạt động bằng cách trước tiên phân chia hình ảnh đầu vào thành một lưới các ô (grid of cells), trong đó mỗi ô chịu trách nhiệm dự đoán các bounding boxes nếu tâm của nó nằm trong ô. Mỗi grid cell (tức 1 ô bất kì nằm trong lưới ô) dự đoán các bounding boxes được xác định dựa trên tọa độ x, y (thông thường là tọa độ tâm, một số phiên bản là tọa độ góc trên cùng bên trái) và chiều rộng (width) và chiều cao (height) và độ tin cậy (confidence) về khả năng chứa vật thể bên trong. Ngoài ra các dự đoán nhãn cũng được thực hiện trên mỗi một bonding box [5].

Ví dụ: một hình ảnh có thể được chia thành lưới 7 × 7 và mỗi ô trong lưới có thể dự đoán 2 bounding box, kết quả trả về 98 bounding box được đề xuất. Sau đó, một sơ đồ xác suất nhãn (gọi là class probability map) với các confidence được kết hợp thành một tợp hợp bounding box cuối cùng và các nhãn. Hình ảnh được lấy từ bài báo dưới đây tóm tắt hai kết quả đầu ra của mô hình.



*Hình 1.2.2.1: Hình ảnh mô tả ký thuật YOLO (2015)*

Các bước xử lý trong mô hình YOLO (hình ảnh trích xuất từ bài báo gốc). Đầu tiên mô hình chia hình ảnh thành một grid search kích thước **S** \* **S**. Trên mỗi một grid cell ta dự báo một số lượng **B** bounding boxes và confidence cho những boxes này và phân phối xác suất của **C** classes. Như vậy output các dự báo là một tensor kích thước **S** \* **S** \* **(B** \* **5** + **C)** . Giá trị 5 là các tham số của offsets của bounding box gồm x, y ,w, h và confidence. $C$ là số lượng tham số của phân phối xác suất [5].

1.2.2.2. YOLOv2 (2016) và YOLOv3 (2018)

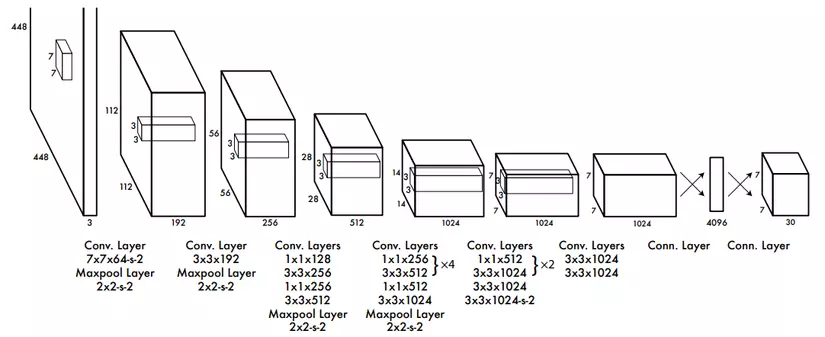
Mô hình YOLOv2 được Joseph Redmon và Ali Farhadi cập nhật nhằm cải thiện hơn nữa hiệu suất trong bài báo năm 2016 có tựa đề là [YOLO9000: Better, Faster, Stronger](https://arxiv.org/abs/1612.08242).

Mặc dù biến thể của YOLO được gọi là YOLOv2, một instance của mô hình theo như mô tả đã được đào tạo trên hai bộ dữ liệu nhận dạng đối tượng, và có khả năng dự đoán lên tới 9000 loại đối tượng khác nhau, do đó được đặt tên là YOLO9000. Với con số này thì mô hình này đã tiến xa hơn rất nhiều so với mọi mô hình trước đó về số lượng các loại đối tượng có khả năng phát hiện [6].

Một số thay đổi về huấn luyện và kiến trúc đã được thực hiện, chẳng hạn như việc sử dụng batch normalization cho hàng loạt và hình ảnh đầu vào phân giải cao.

Giống như Faster R-CNN, mô hình YOLOv2 sử dụng anchor boxes, bounding box được xác định trước với hình dạng và kích thước hợp lý được tùy chỉnh trong quá trình huấn luyện. Sự lựa chọn các bounding boxes cho hình ảnh được xử lý trước bằng cách sử dụng thuật toán phân cụm k-mean trên tập dữ liệu huấn luyện [6].

Điều quan trọng, các predicted bounding box được tinh chỉnh để cho phép các thay đổi nhỏ có tác động ít hơn đến các dự đoán, dẫn đến mô hình ổn định hơn. Thay vì dự đoán trực tiếp vị trí và kích thước, các offsets (tức tọa độ tâm, chiều dài và chiều rộng) được dự đoán để di chuyển và định hình lại các pre-defined anchor boxes tại mỗi một grid cell thông qua hàm logistic [6].



*Hình 1.2.2.2: Hình ảnh mô tả kỹ thuật YOLO*

Sơ đồ giúp tạo prior bounding box có chiều rộng **p**w và chiều cao **p**h đã xác định từ grid cell có tọa độ (cx, cy). Khi đó tọa độ tâm (bx, by) được tính theo mức độ tịnh tiến hàm sigmoid. Đồng thời, chiều rộng và chiều cao (bw, bh) được tính như công thức scale số mũ của cơ số tự nhiên **e** [6].

Ưu điểm của yolo:

Một trong những ưu điểm chính của YOLO là tốc độ suy luận nhanh, cho phép nó xử lý hình ảnh theo thời gian thực. Nó rất phù hợp cho các ứng dụng như giám sát bằng video, ô tô tự lái và thực tế tăng cường.

Nhược điểm:

1. Độ chính xác thấp hơn trong các trường hợp đối tượng nhỏ hoặc dày đặc

YOLO sử dụng các ô lưới (grid cells) để chia ảnh và dự đoán vị trí đối tượng. Điều này khiến mô hình gặp khó khăn khi đối tượng:

Có kích thước nhỏ và rơi vào nhiều ô lưới khác nhau.

Nằm gần nhau hoặc chồng chéo, đặc biệt trong các cảnh đông đúc.

2. Khả năng nhận diện các đối tượng bất đối xứng hoặc phức tạp

YOLO gặp hạn chế trong việc nhận diện các đối tượng có hình dạng không chuẩn, phức tạp hoặc đối tượng bị biến dạng.

3. Cân bằng giữa tốc độ và độ chính xác

Mặc dù nhanh, YOLO thường không đạt độ chính xác cao bằng các mô hình khác như Faster R-CNN trong các tác vụ yêu cầu độ chính xác cực kỳ cao.

4. Phân chia ô lưới làm giảm hiệu quả

Một đối tượng lớn có thể rơi vào một ô lưới duy nhất, trong khi các đối tượng nhỏ có thể rơi vào nhiều ô lưới, dẫn đến dự đoán không chính xác.

Việc phân chia cố định (fixed grid) không phù hợp với các ảnh có bố cục hoặc mật độ đối tượng đa dạng.

5. Khó nhận diện đối tượng ẩn hoặc bị che khuất

YOLO không được thiết kế tối ưu để xử lý các đối tượng bị che khuất một phần, do mô hình tập trung vào các đặc trưng dễ nhận diện.

6. Không linh hoạt trong việc xử lý độ phân giải và tỉ lệ khung hình

YOLO yêu cầu một kích thước đầu vào cố định. Việc chuyển đổi kích thước ảnh để phù hợp với mô hình có thể làm mất thông tin chi tiết hoặc làm méo hình ảnh.

7. Hạn chế trong việc học các đặc trưng nhỏ và hiếm

Mô hình thường bỏ qua các đối tượng hoặc chi tiết nhỏ, đặc biệt nếu chúng không thường xuyên xuất hiện trong tập dữ liệu huấn luyện.

## **1.3. Ngôn ngữ lập trình**

### 1.3.1.Python

là một ngôn ngữ lập trình hướng đối tượng, cấp cao và vô cùng mạnh mẽ. Điều đặc biệt ở ngôn ngữ này là sự đa năng, đa công dụng. Mang các tính năng ưu việt như tạo kiểu dữ liệu tự động và cơ chế cung cấp bộ nhớ tự động. Ngoài ra, Python còn được coi là một ngôn ngữ lập trình có cấu trúc dữ liệu cao cấp và mạnh mẽ nhất.

Các tính năng chính thể hiện sự ưu việt của Python có thể liệt kê như:

* Cực kỳ đơn giản và dễ tiếp thu cũng như thực hành: sự đơn giản của ngôn ngữ này thể hiện ở các câu lệnh dễ học, dễ hiểu. Python biến những “cú pháp code khó nhằn” trở nên dễ dàng hơn bằng những giải pháp.
* Hoàn toàn miễn phí và là mã nguồn mở: nhà lập trình có thể thoải mái di chuyển Python thậm chí sử dụng nó với mục đích thương mại hóa. Với đặc trưng là một ngôn ngữ lập trình mã nguồn mở. Bạn hoàn toàn có thể thay đổi các mã nguồn mở của Python ở bất cứ các ứng dụng phần mềm nào.
* Là một ngôn ngữ di động: di động có nghĩa là nó rất dễ di chuyển từ nền tảng này sang các nền tảng khác. Đồng thời, bạn có thể chạy chương trình của Python ở mọi nền tảng mà không sợ có bất kỳ những thay đổi nào ảnh hưởng.
* Python là ngôn ngữ phiên dịch cao cấp:khi chạy chương trình Python trên bất cứ hệ điều hành hay mạng Internet nào. Python có thể hoàn toàn tự động dịch và chạy theo đúng chương trình dữ liệu của máy tính đó.



**Ưu Điểm:**

* Python có cấu trúc đơn giản, rõ ràng và rất dễ học. Đây được coi là một trong những ngôn ngữ nên học cho những người mới bắt đầu tiếp cận đến lập trình.
* Python có mặt trên tất cả các hệ điều hành phổ biến như: Windows, MacOS, MS-DOS, Unix,… đây cũng thể hiện được sự đa năng của ngôn ngữ này.
* Tương thích với thư viện lớn nhất hiện nay với 400 triệu người sử dụng như data mining Scikit-learn, Pandas…
* Tốc độ xử lý cũng là một trong những lí do tạo nên ưu điểm của Python. Ngôn ngữ này có thể tạo ra những script siêu nhỏ tới những phần mềm cực lớn như Blender 3D.

**Nhược Điểm:**

* Các cấu trúc của Python đòi hỏi nhiều không gian bộ nhớ hơn, ngôn ngữ này không thể phát huy hết tác dụng trong điều kiện bộ nhớ hạn chế.
* Python không phải là ngôn ngữ được ưu tiên để phát triển ứng dụng di động hay trò chơi do tiêu tốn nhiều bộ nhớ hơn và tốc độ xử lý chậm so với ngôn ngữ khác.
* Rất khó để chạy các bài kiểm tra trên mã được viết bằng Python, các lỗi trên này thường xuất hiện trong thời gian chạy, điều này khiến việc kiểm tra trở nên khó khăn hơn.

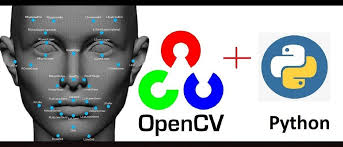
### 1.3.2.Thư viện sử dụng:

OpenCV là tên viết tắt của open source computer vision library – có thể được hiểu là một thư viện nguồn mở cho máy tính. Cụ thể hơn OpenCV là kho lưu trữ các mã nguồn mở được dùng để xử lý hình ảnh, phát triển các ứng dụng đồ họa trong thời gian thực.

OpenCV cho phép cải thiện tốc độ của CPU khi thực hiện các hoạt động real time. Nó còn cung cấp một số lượng lớn các mã xử lý phục vụ cho quy trình của thị giác máy tính hay các learning machine khác.

Thư viện OpenCV được phát hành với giấy phép BDS. Do đó các dịch vụ nó cung cấp là hoàn toàn miễn phí và được hạn chế tối đa các rào cản thông thường. Cụ thể, bạn được phép sử dụng phần mềm này cho cả hoạt động thương mại lẫn phi thương mại. OpenCV sở hữu giao diện thiên thiện với mọi loại ngôn ngữ lập trình, ví dụ như C++, C, Python hay Java… Ngoài ra, nó cũng dễ dàng tương thích với các hệ điều hành khác nhau, bao gồm từ Windows, Linux, Mac OS, iOS cho đến cả Android.

Kể từ lần đầu xuất hiện từ năm 1999, giờ đây OpenCV đã sở hữu đội ngũ người dùng hùng hậu, con số ước tính có thể lên tới 47.000 người. Tất cả là nhờ những ưu điểm vượt trội của OpenCV.



**Các module được dùng trong OpenCV**

OpenCV có cấu trúc module, tức là nó bao gồm cả những thư viện liên kết tĩnh lẫn thư viện liên kết động. Nắm rõ các module của OpenCV sẽ giúp bạn đọc hoàn toàn thấu hiểu OpenCV là gì.

* Core functionality (core): Module này sở hữu cơ chế rất nhỏ gọn. Nó được dùng để định hình các cấu trúc của cơ sở dữ liệu cơ bản, bao gồm cả những mảng đa chiều. Ngoài ra nó còn xác định các chức năng của những module đi kèm khác nữa.
* Image Processing (imgproc): Đây là module được dùng cho quá trình xử lý hình ảnh. Nó cho phép người dùng thực hiện các hoạt động như lọc hình ảnh tuyến tính và phi tuyến, thực hiện phép biến hình, thay đổi không gian màu, xây dựng biểu đồ và rất nhiều thao tác khác liên quan.
* Video Analysis (video): Giống như tên gọi của nó, module này cho phép phân tích các video. Kết quả được trả về bao gồm các ước tính chuyển động, thực hiện tách nền và các phép toán theo dõi vật thể.
* Camera Calibration and 3D Reconstruction (calib3d): Module này cung cấp các thuật toán hình học đa chiều cơ bản và hiệu chuẩn máy ảnh single và stereo. Ngoài ra nó còn đưa ra các dự đoán kiểu dáng của đối tượng và sử dụng thuật toán thư tín âm thanh nổi cùng các yếu tố tái tạo 3D.
* 2D Features Framework (features2d): Module này giúp phát hiện các tính năng nổi trội của bộ nhận diện, bộ truy xuất thông số và thông số đối chọi.
* Ngoài ra còn có rất nhiều module khác với đa dạng tính năng, ví dụ như: FLANN, Google test wrapper

**Ứng dụng của OpenCV**

* OpenCV giúp định vị hình ảnh
* OpenCV được cho là một phần mềm đa nhiệm. Nó được ứng dụng trong rất nhiều trường hợp khác nhau. Ví dụ, ta sẽ nói về các phần mềm định vị, bản đồ nói chung. Hẳn rằng trong chúng ta ai cũng đã có ít nhất một lần cần sử dụng đến các map online đúng không. Bạn sử dụng các map để tìm đường, tra cứu tình hình giao thông hoặc đơn giản là xem xét các hình ảnh thực tế của địa điểm cần đến. Những lúc như vậy, OpenCV đóng vai trò là nhà cung cấp dữ liệu hình ảnh cho các app về Map này. OpenCV sẽ đem đến cho người dùng hình ảnh về đường phố hay các căn nhà, con người xung quanh địa điểm được chỉ định.
* OpenCV còn được dùng để khởi tạo ra những hình ảnh 3 chiều phức tạp. Hoạt động này rất được yêu thích, nhất là trong thời đại trí tuệ nhân tạo AI phát triển như thế này.
* Đối với các công nghệ hiện đại, OpenCV cũng là một yếu tố không thể thiếu. Tất cả những ứng dụng công nghệ như robot, xe tự lái, bảng cảm ứng thông minh… đều có sự góp mặt của OpenCV trong khâu xử lý hình ảnh. Ví dụ gần gũi nhất trong cuộc sống có thể kể đến hệ thống mở khóa điện thoại bằng cách nhận diện khuôn mặt người dùng.

# **Chương 2: Xây dựng hệ thống nhận dạng**

## **2.1. Mô tả bài toán:**

### 2.1.1 Input và output:

a, Input:

* Hình ảnh người:

+ Hình ảnh có thể là 1 ảnh khuôn mặt người có một người hoặc nhiều người, ảnh có thể là ảnh chân dung hoặc ảnh toàn cảnh, ảnh một người hoặc một nhóm người

+ Hình ảnh thu được từ camera của laptop, camera an ninh,....

* Yêu cầu xử lý:
* Khoanh vùng được khuôn mặt đối tượng nhận dạng

+ Dự đoán tuổi( biểu diễn bằng số nguyên)

+ Dự đoán giới tính( biểu diễn dưới dạng nhãn)

b, Output:

* Xác định vị trí khuôn mặt :
* Khoanh vùng vị trí khuôn mặt trên ảnh hoặc trên màn hình hiển thị. Thông tin này có thể được thể hiện bằng hình chữ nhật (bounding box).
* Gán nhãn tuổi và giới tính nhận dạng được:

+ Tuổi: Đưa ra số nguyên tương ứng với dự đoán tuổi và hiển thị lên trên màn hình kết quả dự đoán.

+ Giới tính: Gán nhãn giới tính cho đối tượng nhận dạng và hiển thị lên trên màn hình kết quả.

### 2.1.2. Phương pháp áp dụng:

*2.1.2.1. Bài toán phát hiện khuôn mặt*

Xác định các vị trí khuôn mặt trong ảnh và cắt vùng chứa khuôn mặt.

Sử dụng CNN để phát hiện khuôn mặt:

* **Lớp Convolutional**: Áp dụng các bộ lọc để phát hiện các đặc trưng như mắt, mũi, miệng và hình dáng khuôn mặt.
* **Lớp Pooling**: Giảm kích thước của feature map để tập trung vào các đặc trưng quan trọng nhất.
* **Lớp Fully Connected**: Xác định vị trí khuôn mặt bằng cách dựa vào các đặc trưng đã phát hiện và kết hợp chúng để tạo ra bounding box.

Đầu ra của mô hình:

* Danh sách các bounding box (tọa độ của khuôn mặt) và độ tin cậy (confidence score) cho mỗi khuôn mặt được phát hiện.

*2.1.2.2. Bài toán phân lớp tuổi*

Đưa ra ước lượng tuổi của một người từ bức ảnh chụp khuôn mặt của họ.

Sử dụng khuôn mặt đã được nhận dạng để đoán tuổi:

* **Lớp Convolutional**: Phát hiện các đặc trưng của khuôn mặt liên quan đến tuổi, như nếp nhăn, hình dáng khuôn mặt và kết cấu da.
* **Lớp Pooling**: Giảm kích thước của feature map để giữ lại các đặc trưng quan trọng nhất liên quan đến tuổi.
* **Lớp Fully Connected**: Dự đoán tuổi bằng cách phân loại xác suất khuôn mặt thuộc các nhóm tuổi khác nhau.

Đầu ra của mô hình:

* Nhóm tuổi có xác suất cao nhất sẽ được chọn là nhóm tuổi của đối tượng.

*2.1.2.3.Bài toán phân lớp giới tính*

Sử dụng khuôn mặt đã được nhận dạng để dự đoán giới tính và phân loại xác suất khuôn mặt thuộc các nhóm thuộc tính

Sử dụng khuôn mặt đã được nhận dạng để dự đoán giới tính:

* **Lớp Convolutional**: Phát hiện các đặc trưng của khuôn mặt liên quan đến giới tính, như cấu trúc xương và hình dáng khuôn mặt.
* **Lớp Pooling**: Giảm kích thước của feature map để giữ lại các đặc trưng quan trọng nhất liên quan đến giới tính.
* **Lớp Fully Connected**: Dự đoán giới tính bằng cách phân loại xác suất khuôn mặt thuộc các nhóm giới tính khác nhau.

Đầu ra của mô hình:

* Nhóm giới tính có xác suất cao nhất sẽ được chọn là nhóm giới tính của đối tượng.

## **2.2. Xây dựng hệ thống**

****

Bài toán nhận dạng khuôn mặt và gán nhãn tuổi, giới tính là một vấn đề quan trọng trong lĩnh vực thị giác máy tính, với mục tiêu tự động phân tích hình ảnh khuôn mặt để xác định danh tính và các thông tin liên quan. Bài toán được chia thành ba phần chính:

**Phát hiện khuôn mặt:**

* Mục tiêu là xác định vị trí của các khuôn mặt xuất hiện trong ảnh hoặc video. Điều này bao gồm việc phát hiện khuôn mặt với các điều kiện như góc chụp, ánh sáng không đồng đều, hoặc hình ảnh chứa nhiều khuôn mặt.

**Dự đoán tuổi:**

* Sau khi phát hiện khuôn mặt, bài toán yêu cầu ước tính độ tuổi của người trong ảnh. Tuổi có thể được dự đoán dưới dạng giá trị số nguyên (ví dụ: 25 tuổi) hoặc dưới dạng nhóm tuổi (ví dụ: 20–29).

**Dự đoán giới tính:**

* Dựa trên đặc trưng khuôn mặt, hệ thống cần phân loại giới tính của từng khuôn mặt thành hai nhóm chính: nam hoặc nữ.

Để xây dựng hệ thống nhận dạng tuổi và giới tính, ta thực hiện các bước sau:

Bước 1: Phát hiện khuôn mặt (Face Detection)

Mục tiêu: Xác định các vị trí khuôn mặt trong ảnh và cắt vùng chứa khuôn mặt.  
 Phương pháp:

Sử dụng mô hình phát hiện khuôn mặt dựa trên Deep Learning, ví dụ:

Mạng OpenCV DNN với các tệp:opencv\_face\_detector.pbtxt và opencv\_face\_detector\_uint8.pb. opencv\_face\_detector.pbtxt là tệp mô tả cấu trúc của mạng phát hiện khuôn mặt, bao gồm các lớp, kết nối giữa chúng và thông số của từng lớp. Tệp opencv\_face\_detector\_uint8.pb chứa các trọng số đã được huấn luyện sẵn trên một tập dữ liệu lớn

Nguyên tắc hoạt động:

Sử dụng mạng nơ-ron tích chập (CNN) để phát hiện các vùng trong ảnh chứa khuôn mặt.

Đầu ra của mô hình là danh sách các bounding box (tọa độ của khuôn mặt) và độ tin cậy (confidence score).

Yêu cầu đầu vào:

Ảnh đầu vào được chuẩn hóa bằng cách:

Resize về kích thước cố định (ví dụ: 300x300 pixels).

Trừ đi giá trị trung bình của ảnh (MODEL\_MEAN\_VALUES: [104, 117, 123]).

Kết quả đầu ra:

Danh sách các bounding box cho mỗi khuôn mặt được phát hiện, kèm confidence.

Bước 2: Tiền xử lý khuôn mặt

Mục tiêu: Chuẩn hóa khuôn mặt để làm đầu vào cho các mô hình nhận dạng.  
 Phương pháp:

Resize từng khuôn mặt về kích thước phù hợp (ví dụ: 227x227 pixels).

Chuẩn hóa giá trị pixel của ảnh bằng cách:

Trừ giá trị trung bình của dataset (MODEL\_MEAN\_VALUES).

Thay đổi định dạng kênh màu (nếu cần).

Bước 3: Nhận dạng Giới tính (Gender Classification)

Mụctiêu: Xác định giới tính của đối tượng.

Mô hình:

Sử dụng mô hình CNN được định nghĩa bởi:

Kiến trúc mạng: gender\_deploy.prototxt là tệp định nghĩa cấu trúc mạng CNN để phân loại khuôn mặt vào các nhóm giới tính.

Trọng số đã được huấn luyện: gender\_net.caffemodel chứa các trọng số đã được huấn luyện sẵn trên tập dữ liệu lớn về giới tính.

Nguyên lý hoạt động:

+input: Khuôn mặt cắt ra từ ảnh gốc

+output: Xác suất dự đoán cho mỗi lớp.Lớp có xác suất cao nhất là kết quả dự đoán.

Bước 4: Nhận dạng Tuổi (Age Estimation)

Mục tiêu: Dự đoán tuổi hoặc nhóm tuổi của đối tượng.  
Mô hình:

Sử dụng CNN với cấu trúc và trọng số được cung cấp:

Kiến trúc mạng: age\_deploy.prototxt là tệp định nghĩa cấu trúc mạng CNN để phân loại khuôn mặt vào các nhóm tuổi.

Trọng số đã được huấn luyện: age\_net.caffemodel chứa các trọng số của mạng đã được huấn luyện trên tập dữ liệu lớn về tuổi.

Nguyên lý hoạt động:

+input: Khuôn mặt cắt ra từ ảnh gốc

+output: Xác suất dự đoán cho mỗi lớp.Lớp có xác suất cao nhất là kết quả dự đoán.

# 

# **Chương 3: 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:

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).

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.

Xử lý cắt khuôn mặt:

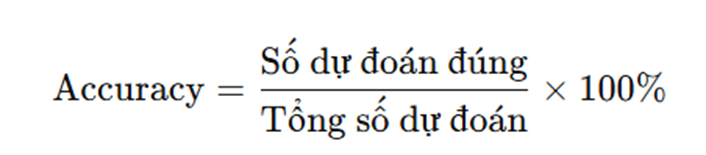
* Sau khi phát hiện khuôn mặt, vùng cắt được mở rộng một chút (thêm 15 pixel mỗi cạnh) để đảm bảo không mất chi tiết.
* Các vùng này được đưa vào mô hình nhận diện giới tính và độ tuổi.

## **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:

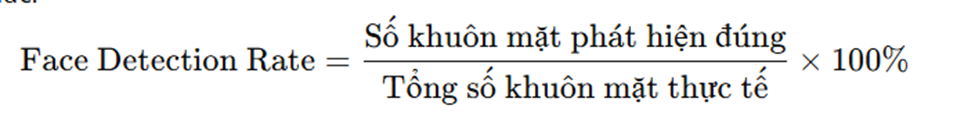
### 3.2.1. Độ chính xác (Accuracy):

* 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:

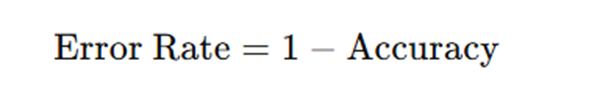


### 3.3.3. Độ tin cậy (Confidence Score):

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:85% (85/100 khuôn mặt được phát hiện chính xác).

Trong 100 khuôn mặt được đưa vào hệ thống, 85 khuôn mặt được phát hiện chính xác.

Nguyên nhân gây lỗi:

Ảnh bị mờ: Do độ phân giải thấp hoặc rung lắc trong lúc chụp.

Góc chụp nghiêng quá lớn: Khuôn mặt không nhìn trực diện hoặc không nằm trong góc phát hiện của thuật toán.

* Nhận diện giới tính:
  + Độ chính xác: 85%.
  + 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: 76%.

Trong 100 khuôn mặt, 76 khuôn mặt được dự đoán đúng độ tuổi trong khoảng mong đợi.

Thuật toán sử dụng các đặc điểm trên khuôn mặt như nếp nhăn, kết cấu da, hình dáng khuôn mặt để ước tính tuổi.

Nguyên nhân gây lỗi:

Khoảng tuổi gần nhau:

Ví dụ: Nhóm tuổi 15-20 và 25-32 có thể bị nhầm lẫn do các đặc điểm khuôn mặt tương đối giống nhau, đặc biệt ở người trẻ tuổi.

Cá nhân hóa:

Một số người có khuôn mặt trẻ hơn hoặc già hơn tuổi thực tế, gây khó khăn cho hệ thống.

**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.
* Trả về các kết quả so sánh nhận dạng như số khuôn mặt nhận dạng được, tỷ lệ nhận dạng đúng, sai  
   Ví dụ:
* Khuôn mặt 1: "Nam, (25-32)"
* Khuôn mặt 2: "Nữ, (15-20)"
* Số khuôn mặt nhận dạng được: 6
* Số nguôn mặt nhận dạng đúng: 5
* Số khuôn mặt nhận dạng sai: 1
* Tỷ lệ nhận dạng đúng: 85,43%
* Tỷ lệ nhận dạng sai: 14,57%

# **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.