**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI: NGHIÊN CỨU TRIỂN KHAI THUẬT TOÁN YOLOV3** **VÀ XÂY DỰNG HỆ THỐNG HƯỚNG DẪN ĐẬU XE Ô TÔ PARKING VISION**

Giảng viên hướng dẫn: ThS. PHẠM THỊ MIÊN

Sinh viên thực hiện : NGUYỄN VĂN HIỆP

Mã sinh viên : 5851071024

Lớp : CÔNG NGHỆ THÔNG TIN

Khoá :58

TP. Hồ Chí Minh, tháng 06 năm 2021

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI:** **NGHIÊN CỨU TRIỂN KHAI THUẬT TOÁN YOLOV3 VÀ XÂY DỰNG HỆ THỐNG HƯỚNG DẪN ĐẬU XE Ô TÔ PARKING VISION**

Giảng viên hướng dẫn: ThS. PHẠM THỊ MIÊN

Sinh viên thực hiện : NGUYỄN VĂN HIỆP

Mã sinh viên : 5851071024

Lớp : CÔNG NGHỆ THÔNG TIN

Khoá :58

TP. Hồ Chí Minh, tháng 06 năm 2021

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI  PHÂN HIỆU TẠI TP. HỒ CHÍ MINH  **BỘ MÔN CÔNG NGHỆ THÔNG TIN**  --------------------------------------- | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**  Độc lập – Tự do - Hạnh phúc  --------------------------------------------------- |

**NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP**

Họ và tên sinh viên: NGUYỄN VĂN HIỆP

MSSV: 5851071024

Chuyên ngành: Công nghệ Thông tin

Lớp: CQ.CNTT.58

1. **Tên đề tài đồ án tốt nghiệp:**

***NGHIÊN CỨU TRIỂN KHAI THUẬT TOÁN YOLOV3 VÀ XÂY DỰNG HỆ THỐNG HƯỚNG DẪN ĐẬU XE Ô TÔ PARKING VISION.***

1. **Mục đích, yêu cầu:**
   1. **Mục đích:**

* Xây dựng website theo dõi các vị trí còn trống và đã đỗ trong bãi đỗ xe và hiển thị lên màn hình số vị trí còn trống, đã đỗ và tổng số lượng chỗ đỗ trong bãi đỗ xe ô tô theo thời gian thực.
* Xây dựng mô hình nhận diện các chỗ còn trống và đã đỗ trong bãi đỗ xe bằng hình ảnh, video, camera.
* Xây dựng website giới thiệu sản phẩm.
* Xây dựng giải pháp cho Hệ Thống Hướng Dẫn Bãi Đỗ Xe (Parking Guidance System) thông minh tiết kiệm, nhanh hơn, tiện dụng, chính xác và có thể mở rộng quy mô sau này.
  1. **Yêu cầu:**
* Tìm hiểu về Thị Giác Máy Tính (Computer Vision) và Học Sâu (Deep Learning).
* Nghiên cứu về xử lý ảnh.
* Nghiên cứu những quy trình trong xử lý ảnh.
* Nghiên cứu về Mạng Neural Tích Chập (Convolution Neural Network) và những ứng dụng của nó trong Deep Learning và Computer Vision.
* Nghiên cứu thuật toán YOLO (You Only Look Once).
* Tìm hiểu một số khái niệm liên quan đến lĩnh lực thống kê học áp dụng vào mô hình Học Sâu.
* Tìm hiểu các nguồn cơ sở dữ liệu hình ảnh cho quá trình huấn luyện mô hình Học Sâu.
* Thu thập dữ liệu hình ảnh về những chỗ trống và đã đỗ trong bãi đỗ xe. Gắn nhãn, tiền xử lý.
* Tìm hiểu về Transfer Learning, và ứng dụng vào huấn luyện mô hình.
* Ứng dụng kiến trúc Darknet huấn luyện mô hình trên tập dữ liệu lớn và tập dữ liệu mô phỏng bằng công cụ Google Colab.
* Ứng dụng thuật toán YOLOv3 (You Only Look Once, Version 3) để phát hiện những vị trí còn trống, đã đỗ trên hình ảnh, video, camera bằng Pytorch.
* Ứng dụng Nicepage thiết kế website giới thiệu sản phẩm và sử dụng Django để xử lý trang web, truyền màn hình đã xử lý lên website.

1. **Nội dung và phạm vi đề tài:** 
   1. **Nội dung đề tài:**

* Giới thiệu và phân biệt các khái niệm liên quan đến Trí Tuệ Nhân Tạo (Artificial Intelligence), Thị Giác Máy Tính (Computer Vision), Học Máy (Machine Learning), Học Sâu (Deep Learning), Khai Phá Dữ Liệu (Data Mining).
* Nghiên cứu và triển khai thuật toán YOLOv3 bằng Pytorch:
* Triển khai mô-đun nhận diện qua hình ảnh.
* Triển khai mô-đun nhận diện qua video/camera.
* Kiểm thử mô hình.
* Nghiên cứu các chỉ số đánh giá mô hình.
* Xây dựng trang web hiển thị màn hình xử lý bằng Django.
* Hiển thị song song màn hình xử lý trên website và trên desktop.

1. **Phạm vi đề tài:**

* Bài toán nhận diện vật thể (Object Detection).
* Giới thiệu họ các thuật toán Mạng Neural Tích chập (Convolution Neural Network) trong nhận diện vật thể.
* Ứng dụng Django để xây dựng website giới thiệu sản phẩm.
* Tích hợp module nhận diện qua camera vào Django để truyền màn hình đã xử lý lên website.

1. **Công nghệ, công cụ và ngôn ngữ lập trình:**
2. **Công nghệ:**

Python, OpenCV, Pytorch, Django, Colab Notebook, CUDA, Darknet.

1. **Công cụ:**

* Một số thư viện mã nguồn mở của Python:

Opencv-python, pandas, numpy, django, torch,...

* Visual Studio Code
* Darknet: Open Source Neural Networks
* Google Colab

1. **Ngôn ngữ lập trình:** Python
2. **Các kết quả chính dự kiến sẽ đạt được và ứng dụng:**

* Sử dụng camera tiến hành phát hiện các vị trí còn trống và đã đỗ trong thời gian thực. Hiển thị lên màn hình vị trí còn trống và đã đỗ, đếm những vị trí còn trống, đã đỗ, tổng các vị trí hiện có.
* Sử dụng Django để xây dựng website.
* Mô phỏng mô hình trực tiếp.
* Hoàn chỉnh cuốn báo cáo đề tài.
* Nắm được kiến trúc thuật toán YOLOv3 và có thể ứng dụng vào mọi đề tài liên quan.
* Nắm được các ưu, nhược điểm của thuật toán và các phương pháp tối ưu cho thuật toán.
* Nắm được những quy trình trong huấn luyện và kiểm tra mô hình trong các mô hình Deep Learning.

1. **Giáo viên và cán bộ hướng dẫn**

Họ tên: PHẠM THỊ MIÊN

Đơn vị công tác: Bộ môn Công Nghệ Thông Tin – Trường Đại học Giao thông Vận tải phân hiệu tại TP HCM

Điện thoại: 0961170638 Email:

|  |  |
| --- | --- |
| **Ngày … tháng 03 năm 2021**  **BM Công Nghệ Thông Tin** | **Đã giao nhiệm vụ TKTN**  **Giáo viên hướng dẫn** |
|  | **ThS. Phạm Thị Miên** |

Đã nhận nhiệm vụ TKTN

Sinh viên: NGUYỄN VĂN HIỆP Ký tên:

Điện thoại: 0973550322

Email: [nguyenvanhiepcmg@gmail.com](mailto:nguyenvanhiepcmg@gmail.com)

**LỜI CẢM ƠN**

Qua thời gian học tập và rèn luyện tại trường Trường Đại học Giao thông Vận tải phân hiệu tại TP HCM, đến nay chúng em đã kết thúc khoá học 4 năm và hoàn thành đồ án tốt nghiệp. Trong thời gian học tập tại trường để có được kết quả hiện tại em xin chân thành cảm ơn:

Cảm ơn tập thể các thầy cô giáo Bộ môn Công Nghệ Thông Tin và các thầy cô thỉnh giảng đã giảng dạy, quan tâm và tạo điều kiện thuận lợi để chúng em học tập rèn luyện trong suốt thời gian qua, giúp chúng em trang bị những kiến thức, kỹ năng cần thiết cho công việc thực tế sau này. Cảm ơn thầy cô giáo Bộ môn cũng như Ban Giám Hiệu đã cho phép em thực hiện đề tài tốt nghiệp: ***NGHIÊN CỨU TRIỂN KHAI THUẬT TOÁN YOLOV3 VÀ XÂY DỰNG HỆ THỐNG HƯỚNG DẪN ĐẬU XE Ô TÔ PARKING VISION***.

Và cảm ơn thạc sĩ Phạm Thị Miên đã luôn quan tâm nhiệt tình hướng dẫn, giúp đỡ em trong quá trình thực hiện đồ án tốt nghiệp. Cô cũng luôn nhắc nhở, động viên mỗi khi em gặp khó khăn, nhờ vậy mà em đã hoàn thành tốt đồ án tốt nghiệp của mình đúng thời hạn được giao. Nếu không có những lời hướng dẫn, dạy bảo của cô thì em nghĩ bài báo cáo này của em sẽ rất khó có thể hoàn thiện được.

Em cũng xin gửi lời cảm ơn tới gia đình, bạn bè, những người đã động viên, giúp đỡ em rất nhiều trong thời gian học tập và làm đồ án tốt nghiệp.

Mặc dù đã cố gắng nỗ lực học hỏi không ngừng để hoàn thành đề tài, vậy nhưng thời gian thực hiện đồ án có hạn, kiến thức của em còn hạn chế. Do vậy, không tránh khỏi những thiếu sót, em rất mong nhận được những ý kiến đóng góp quý báu của thầy cô trong hội đồng bảo vệ đồ án tốt nghiệp để kiến thức của em được hoàn thiện hơn.

Đồng thời em xin cam đoan rằng nội dung đồ án của chính em nghiên cứu xây dựng nên, nếu có nội dung tham khảo đều được trích dẫn cụ thể, rõ ràng.

***TP. Hồ Chí Minh, ngày 11 tháng 06 năm 2021***

**Sinh viên thực hiện**

**Nguyễn Văn Hiệp**

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

|  |
| --- |
| ***Tp. Hồ Chí Minh, ngày … tháng 06 năm 2021***  **Giáo viên hướng dẫn**  **ThS. Phạm Thị Miên** |

**MỤC LỤC**

**DANH MỤC THUẬT NGỮ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **THUẬT NGỮ** | **Ý NGHĨA TIẾNG VIỆT** | **TỪ VIẾT TẮT** | **GHI CHÚ** |
| 1 | You Only Look Once | Bạn Chỉ Nhìn Một Lần | YOLO |  |
| 2 | You Only Look Once Version 2 | Bạn Chỉ Nhìn Một Lần Phiên bản 2 | YOLOv2 |  |
| 3 | You Only Look Once Version 3 | Bạn Chỉ Nhìn Một Lần Phiên bản 3 | YOLOv3 |  |

**DANH MỤC CÁC BẢNG**

**DANH MỤC HÌNH ẢNH**

[Hình 1. Chiếc bán tải bị tạt sơn khi đỗ trên vỉa hè. Ảnh Internet 1](#_Toc74723900)

[Hình 2. Bãi đỗ xe truyền thống tại siêu thị Emart Gò Vấp TP. HCM. Ảnh Internet 1](#_Toc74723901)

[Hình 3. Bãi đỗ xe tự động tại Đà Nẵng. Ảnh Internet 1](#_Toc74723902)

[Hình 4. Minh họa hướng dẫn đỗ xe thủ công. Ảnh Internet 1](#_Toc74723903)

[Hình 5. Minh họa cho hệ thống hướng dẫn bằng cảm biến 1](#_Toc74723904)

[Hình 6. Minh họa kích thước ô tô tiêu chuẩn 1](#_Toc74723905)

[Hình 7. Tiêu chuẩn kích thước bãi đỗ xe thông dụng 1](#_Toc74723906)

[Hình 8. Tiêu chuẩn thiết kế bãi đỗ xe 45 độ 1](#_Toc74723907)

[Hình 9. Tiêu chuẩn thiết kế bãi đỗ xe song song 1](#_Toc74723908)

[Hình 10. Minh họa Camera Analog 1](#_Toc74723909)

[Hình 11. Minh họa Camera Wifi 1](#_Toc74723910)

[Hình 1.1 Mối quan hệ giữa AI, Machine Learning và Deep Learning. **Error! Bookmark not defined.**](#_Toc74723911)

[Hình 1.2 Mối quan hệ giữa AI, Machine Learning và Deep Learning. **Error! Bookmark not defined.**](#_Toc74723912)

[Hình 1.3 Lựa chọn thuật toán phù hợp trong Machine Learning. **Error! Bookmark not defined.**](#_Toc74723913)

[Hình 3.1 Dự liệu sau khi lấy trên diễn đàn 1](#_Toc74723914)

[Hình 3.2 Dữ liệu sau khi được chuẩn hóa 1](#_Toc74723915)

[Hình 3.3 Form phân loại câu hỏi sinh viên 1](#_Toc74723916)

[Hình 3.4 Quy trình hóa phân loại văn bản 1](#_Toc74723917)

[Hình 3.5 Các tag phân loại trên diễn đàn 1](#_Toc74723918)

[Hình 3.6 Quá trình tạo mô hình véc- tơ và ma trận câu hỏi 1](#_Toc74723919)

[Hình 3.7 Quá trình chọn câu hỏi-câu trả lời tương tự trên diễn đàn 1](#_Toc74723920)

[Hình 3.8 Quá trình phân tag cho câu hỏi 1](#_Toc74723921)

[Hình 3.9 Tổ chức tập dữ liệu train và test 1](#_Toc74723922)

[Hình 3.10 Kết quả sau khi train tập dữ liệu trên 1](#_Toc74723923)

[Hình 3.11 Công thức dự đoán điểm 1](#_Toc74723924)

[Hình 3.12 Quản lý công thức dự đoán điểm 1](#_Toc74723925)

[Hình 3.13 Sơ đồ usecase tổng quan của hệ thống 1](#_Toc74723926)

[Hình 3.14 Sơ đồ hoạt động đăng nhập 1](#_Toc74723927)

[Hình 3.15 Sơ đồ hoạt động quản lý dự đoán điểm 1](#_Toc74723928)

[Hình 3.16 Sơ đồ hoạt động dự đoán điểm 1](#_Toc74723929)

[Hình 3.17 Sơ đồ hoạt động quản lý người dùng 1](#_Toc74723930)

[Hình 3.18 Sơ đồ hoạt động dự đoán điểm 1](#_Toc74723931)

[Hình 3.19 Sơ đồ hoạt động hỗ trợ trực tuyến 1](#_Toc74723932)

[Hình 3.20 Sơ đồ hoạt động quản lý diễn đàn 1](#_Toc74723933)

[Hình 3.21 Sơ đồ hoạt động quản lý báo cáo thống kê 1](#_Toc74723934)

[Hình 3.22 Giao diện tạo tài khoản đăng nhập 1](#_Toc74723935)

[Hình 3.23 Giao diện đăng nhập 1](#_Toc74723936)

[Hình 3.24 Giao diện trang chủ của website 1](#_Toc74723937)

[Hình 3.25 Giao diện đặt câu hỏi trên diễn đàn 1](#_Toc74723938)

[Hình 3.26 Giao diện trả lời câu hỏi 1](#_Toc74723939)

[Hình 3.27 Các chức năng thao tác với câu hỏi 1](#_Toc74723940)

[Hình 3.28 Giao diện chức năng tìm kiếm câu hỏi 1](#_Toc74723941)

[Hình 3.29 Giao diện chức năng chat trực tuyến 1](#_Toc74723942)

[Hình 3.30 Giao diện người dùng sử dụng chức năng chat trực tuyến 1](#_Toc74723943)

[Hình 3.31 Giao diện chức năng dự đoán điểm 1](#_Toc74723944)

[Hình 3.32 Giao diện bảng điểm sau khi người dùng chọn chức năng dự đoán điểm 1](#_Toc74723945)

[Hình 3.33 Giao diện nhóm chức năng quản lý 1](#_Toc74723946)

[Hình 3.34 Giao diện quản lý người dùng 1](#_Toc74723947)

[Hình 3.35 Giao diện hiện thị người dùng truy cập trang web 1](#_Toc74723948)

[Hình 3.36 Giao diện hiện thị người dùng truy cập trang web 1](#_Toc74723949)

[Hình 3.37 Giao diện theo dõi lịch sử chat 1](#_Toc74723950)

[Hình 3.38 Giao diện quản lý dự đoán điểm 1](#_Toc74723951)

# **TỔNG QUAN**

## **Sơ lược về Computer Vision và sự hỗ trợ của Deep Learning**

### ***Thị Giác Máy Tính (Computer Vision)***

Thị Giác Máy Tính (Computer Vision) đề cập đến toàn bộ quá trình mô phỏng tầm nhìn của con người trong một bộ máy phi sinh học. Điều này bao gồm việc chụp ảnh ban đầu, phát hiện và nhận dạng đối tượng, nhận biết bối cảnh tạm thời giữa các cảnh và phát triển sự hiểu biết ở mức độ cao về những gì đang xảy ra trong khoảng thời gian thích hợp. Trong thực tế, một hệ thống cung cấp khả năng Thị Giác Máy Tính đáng tin cậy, chính xác và trong thời gian thực là một vấn đề đầy thách thức vẫn chưa được phát triển một cách đầy đủ.

Khi các hệ thống này được phát triển hoàn chỉnh, sẽ có vô số ứng dụng dựa vào Thị Giác Máy Tính như một thành phần chính. Những ví dụ điển hình đó là xe hơi tự lái, robot tự động, máy bay không người lái, thiết bị hình ảnh y tế thông minh hỗ trợ phẫu thuật và cấy ghép phẫu thuật phục hồi thị lực của con người.

Có thể thấy Thị Giác Máy Tính mang nhiều hứa hẹn rất lớn trong tương lai, tuy vậy nó là một hệ thống phức tạp và luôn là thách thức đối với các hệ thống máy tính. Một phần của sự phức tạp là do thực tế Thị Giác Máy Tính không phải là một nhiệm vụ duy nhất. Thay vào đó, nó là một chuỗi các nhiệm vụ không đơn giản mà mỗi yêu cầu sử dụng các thuật toán phức tạp và đủ sức mạnh tính toán để hoạt động trong thời gian thực. Ở cấp độ cao, các tác vụ phụ của Thị Giác Máy Tính là phát hiện và phân đoạn đối tượng, phân loại hình ảnh, theo dõi đối tượng, gắn nhãn hình ảnh với các mô tả có ý nghĩa (ví dụ như chú thích hình ảnh) và cuối cùng, hiểu ý nghĩa của toàn bộ bối cảnh.

### ***Đóng góp từ Deep Learning***

Mặc dù vẫn còn những trở ngại đáng kể trong con đường phát triển của Thị Giác Máy Tính đến “cấp độ con người”, các hệ thống Deep Learning đã đạt được tiến bộ đáng kể trong việc xử lý một số nhiệm vụ phụ có liên quan. Lý do cho sự thành công này một phần dựa trên trách nhiệm bổ sung được giao cho các hệ thống Deep Learning. Điều hợp lý để nói rằng sự khác biệt lớn nhất với các hệ thống Deep Learning là chúng không còn cần phải được lập trình để tìm kiếm các đặc điểm cụ thể. Thay vì tìm kiếm các đặc điểm cụ thể bằng thuật toán được lập trình cẩn thận, các mạng lưới Neural bên trong các hệ thống Deep Learning được đào tạo để có khả năng tự học. Ví dụ: nếu ô tô trong hình ảnh bị phân loại sai thành xe máy thì ta không tinh chỉnh các tham số hoặc viết lại thuật toán. Thay vào đó, ta tiếp tục đào tạo cho đến khi hệ thống làm cho đúng.

Với sức mạnh tính toán tăng lên được cung cấp bởi các hệ thống Deep Learning hiện đại, có sự tiến bộ ổn định và đáng chú ý hướng tới điểm mà một máy tính sẽ có thể nhận ra và phản ứng với mọi thứ mà nó nhìn thấy.

### ***Một số ứng dụng chính***

* Phân loại hình ảnh
* Phân loại hình ảnh cùng với việc bản địa hóa
* Phát hiện đối tượng
* Tái thiết hình ảnh
* Theo dõi đối tượng

Những tiến bộ trong hệ thống Deep Learning và sức mạnh tính toán đã giúp cải thiện tốc độ, độ chính xác và độ tin cậy tổng thể của hệ thống Thị Giác Máy Tính. Khi các mô hình Deep Learning cải thiện và sức mạnh tính toán trở nên dễ dàng hơn, chúng ta sẽ tiếp tục đạt được những tiến bộ và sự ổn định đối với các hệ thống tự vận hành có thể thực sự nắm bắt và phản ứng với những gì chúng cảm nhận.

**Đặt vấn đề**

***Thực trạng***

Cùng với sự phát triển vượt bậc về kinh tế xã hội, nhu cầu của con người ngày càng được nâng cao, do đó nhu cầu lưu thông hàng hoá và những đòi hỏi về đi lại ngày càng tăng. Có thể nói trong giai đoạn hiện nay, khi xu thế toàn cầu hoá diễn ra ngày càng sôi động, người ta càng nhận thức rõ hơn tầm quan trọng của giao thông vận tải.

Nếu như trên không trung, máy bay là phương tiện chiếm ưu thế, thì trên mặt đất, ô tô và vận tải ô tô lại chiếm ưu thế về năng lực vận chuyển và khả năng cơ động. Ô tô có thể hoạt động trên nhiều dạng địa hình, từ đồng bằng, miền núi đến miền biển, vận chuyển một khối lượng hàng hoá và vận chuyển người nhiều hơn bất cứ loại phương tiện vận tải nào khác.

Theo Sở Giao thông vận tải Hà Nội, trên địa bàn Hà Nội hiện có khoảng 6,9 triệu phương tiện giao thông (ô-tô, xe máy), chưa kể lượng xe ngoại tỉnh ra, vào thành phố hằng ngày. Dự tính, với tốc độ tăng trưởng xe máy 7,66%/năm; ô-tô 16,15%/năm thì đến năm 2025 Hà Nội có 1,3 triệu ô-tô và 7,3 triệu xe máy; năm 2030 có 1,7 triệu ô-tô và 7,7 triệu xe máy. Thế nhưng, điều đáng nói là hệ thống giao thông tĩnh (bãi, điểm đỗ xe công cộng) hiện nay còn quá thiếu và yếu, khiến nhu cầu đỗ xe ngày càng cấp thiết.[[1]](#footnote-1) (*theo báo nhandan.vn số ra ngày 09/03/2021*)

Thực trạng cho ta thấy cùng với sự tăng lên của các phương tiện giao thông chỉ ở Hà Nội, với các thành phố lớn thì cũng đang gặp vấn đề tương tự, ô tô đang có tốc độ tăng trưởng về số lượng khá cao về số lượng, do đó nhu cầu đỗ xe cũng ngày càng cao. Việc tìm được một chỗ đỗ xe trong thời buổi hiện nay cũng đang còn khá khó khăn. Đôi khi, có thể có chỗ trống trong bãi đỗ xe nhưng người sử dụng sẽ không biết nó nằm ở đâu, hoặc là không biết trong bãi có còn chỗ trống hay không. Rất nhiều vụ đỗ xe trước nhà dân, vỉa hè, lòng đường, công viên các nơi không phép… dẫn đến nhiều bất cập, ví dụ tình trạng tắc nghẽn giao thông và mất mỹ quan đô thị gây cản trở đường đi của chủ nhà, thậm chí nguy hiểm cho các phương tiện giao thông khác.



Hình 0.1. Chiếc bán tải bị tạt sơn khi đỗ trên vỉa hè. Ảnh Internet

### ***Các loại hình bãi đỗ xe phổ biến hiện nay***

* **Bãi đỗ xe truyền thống:** Hệ thống đỗ xe ô tô truyền thống có ở hầu hết các gia đình, văn phòng và trung tâm giải trí. Không gian đỗ xe riêng biệt được tạo ra để vào và ra trơn tru để tăng tổ chức và giảm lưu lượng. Chúng đòi hỏi ít năng lượng hơn với yêu cầu năng lượng duy nhất là chiếu sáng. Nhà để xe truyền thống cũng cực kỳ đáng tin cậy ở chỗ chức năng của họ không bị cản trở bởi sự cố mất điện. Nhà để xe truyền thống đang hoạt động suốt ngày đêm, một số bãi đỗ xe truyền thống có không gian hạn chế nên người người sử dụng phải tìm chỗ trống trước khi đỗ xe. Nếu chúng có thiết kế đơn giản, số vốn bỏ ra không lớn. Chiến lược bảo vệ xe hơi cũng có thể được cung cấp bằng cách giao nhiều lô cho các cá nhân cụ thể quản lý.

Hình 0.2. Bãi đỗ xe truyền thống tại siêu thị Emart Gò Vấp TP. HCM. Ảnh Internet

* **Hệ thống bãi đỗ xe tự động:** đòi hỏi ít diện tích hơn, loại bỏ trình điều khiển khỏi quá trình đỗ xe. Nó có thể là hoàn toàn tự động hoặc bán tự động. Chiếc xe phải được lái đến một điểm nhập cảnh nơi người lái và hành khách thoát khỏi xe. Sau đó, nó được di chuyển tự động hoặc bán tự động (với một số hỗ trợ cần thiết) đến không gian được phân bổ của nó. Hệ thống bãi đỗ xe tự động tối đa hóa không gian hạn chế, một lợi thế trong khu vực không gian hạn chế có sẵn.



Hình 0.3. Bãi đỗ xe tự động tại Đà Nẵng. Ảnh Internet

### ***Hệ Thống Hướng Dẫn Bãi Đỗ Xe (Parking Guidance System - PGS)***

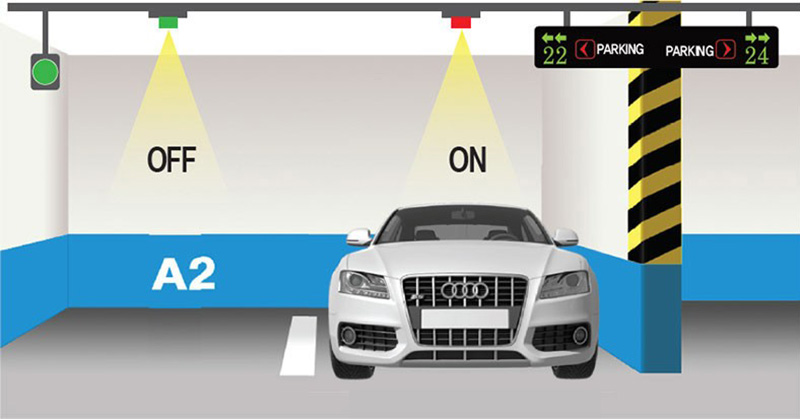
Cho dù là bãi đỗ xe truyền thống hay tự động thì trong một bãi đỗ xe hoàn chỉnh không thể thiếu hệ thống hướng dẫn đỗ xe (Viết tắt là PGS), các loại hình phổ biến tại Việt Nam hiện nay:

* **Chỉ dẫn thủ công:** Đây là phương pháp chỉ dẫn truyền thống nhất, mọi công việc đều do con người trực tiếp tham gia vào làm, khi đó khi vào chỗ để xe thì người sử dụng sẽ được một nhân viên trong bãi xe hướng dẫn đỗ xe, hoặc nhiều khi đó cũng chính là người ghi vé. Phương pháp này tốn rất nhiều nhân công và thiếu chính xác, ví dụ như một bãi đỗ xe lớn thì nhiều khi họ sẽ không nắm được số lượng chỗ trống còn lại và vị trí nào còn trống trong bãi để chỉ dẫn cho người sử dụng được. Hơn nữa còn mang lại phiền toái khá nhiều cho người sử dụng khi phải mất thời gian tự đi tìm chỗ trống, dễ gây ra va chạm với các xe khác, tốn nhiên liệu tiêu thụ cho xe và tăng mức độ ô nhiễm môi trường, làm tâm trạng người sử dụng trở nên chán nản và bực bội, gây ùn ứ các phương tiện đi vào bãi, có thể làm giảm doanh thu của bãi xe.



Hình 0.4. Minh họa hướng dẫn đỗ xe thủ công. Ảnh Internet

* **Hệ thống hướng dẫn thông minh bằng cảm biến:** Hệ thống hướng dẫn đỗ xe thông minh cung cấp sự tiện lợi trong việc hướng dẫn đỗ xe một cách tự động cho bãi đỗ xe ô tô. Người sử dụng dễ dàng nhận biết vị trí còn trống tại từng khu vực trong bãi đỗ xe khi họ đi vào bãi. Tại mỗi vị trí xe đỗ được gắn một thiết bị siêu âm để phát hiện vị trí đỗ là có xe hay không có xe. Tình trạng của các vị trí báo cáo về trung tâm theo thời gian thực. Máy tính chủ sẽ thu thập tất cả trạng thái của từng vị trí từ bộ siêu âm và điều khiển các bảng quang báo hiển thị để hướng dẫn người sử dụng dễ dàng tìm đến vị trí còn trống để đỗ xe.



Hình 0.5. Minh họa cho hệ thống hướng dẫn bằng cảm biến

Tuy nhiên giá mỗi cảm biến rất đắt đỏ, tốn khá nhiều chi phí triển khai, việc bảo trì cũng diễn ra khá khó khăn và tốn kém khi phải thuê nhân viên bên ngoài về vì yêu cầu tính chuyên môn cao, và giá linh kiện đắt đỏ, hơn nữa không phải lúc nào thì người sử dụng cũng biết là trong bãi còn chỗ trống hay không, đến khi vào nhưng hết chỗ thì cũng gây ra nhiều phiền toái. Hệ thống này chủ yếu được triển khai trong nhà có mái che, hoặc hầm.

### ***Giải pháp mới sử dụng công nghệ Computer Vision và Deep Learning: Hệ thống hướng dẫn đậu xe thông minh Parking Vision***

Để giải quyết những nhược điểm của các phương pháp trên thì giải pháp được đề xuất để giải quyết cũng như là đề tài của đồ án này đó là *nghiên cứu triển khai thuật toán YOLOv3 và xây dựng hệ thống hướng dẫn đậu xe ô tô Parking Vision*. Ta sẽ sử dụng bài toán **nhận diện vật thể** nằm trong lĩnh vực Computer Vision để nhận diện ra các vị trí đã đỗ và còn trống trong bãi, cùng với đó sử dụng công nghệ Deep Learning để huấn luyện và hỗ trợ cho Computer Vision trong việc kiểm tra dự đoán của mô hình. Một trong những sự kết hợp đó là thuật toán You Only Look Once (YOLO), là một mô hình CNN để nhận diện vật thể mà một ưu điểm nổi trội là nhanh hơn nhiều so với những mô hình cũ. Thậm chí có thể chạy tốt trên những thiết bị IOT như Raspberry Pi. Khi sử dụng phương pháp này thì có những ưu điểm sau:

* **Đối với chủ đầu tư:** Giúp chủ đầu tư quản lý bãi giữ xe một cách chuyên nghiệp, khoa học với các công nghệ hiện đại, đồng thời giảm chi phí chủ đầu tư (bao gồm chi phí đầu tư máy chủ, tên miền, màn hình hiển thị và camera đối với giải pháp này), thích hợp triển khai ở những nơi ngoài trời như bãi đỗ xe của siêu thị, trường học, nhà hàng... có thể bao quát tầm nhìn của toàn bộ xe.
* **Đối với người sử dụng:** Thuận tiện cho người sử dụng, có thể nắm bắt số lượng chỗ trống và vị trí chính xác trong bãi, giúp chủ xe giảm thời gian tìm kiếm, tiết kiệm nhiên liệu, giúp tâm trạng thoải mái hơn thay vì tự mình tìm xem vị trí nào còn trống. Nhờ đó có thể biết trước bãi còn chỗ đỗ xe hay không, có thể tự tin đến nơi đỗ xe. Hơn nữa người sử dụng có thể theo dõi tình trạng xe mình ở mọi lúc mọi nơi, sử dụng như một camera an ninh.
* **Đối với người quản lý:** Thuận tiện cho người quản lý bãi xe, giúp cho họ nắm bắt thông tin số lượng chỗ trống hiện tại, chính xác vị trí còn trống để chỉ dẫn cho các tài xế vào bãi, tạo điều kiện thuận lợi cho việc điều hành các phương tiện di chuyển.

**Tình hình nguyên cứu**

Hiện nay ở Việt Nam phương pháp hướng dẫn ô tô dựa vào Computer Vision còn chưa phát triển nhiều, có lẽ vì ngành Computer Vision ở Việt Nam chưa được ứng dụng sâu và rộng, đây là một ngành mới nổi ở Việt Nam và có nhiều triển vọng phát triển, hiện tại chỉ được áp dụng nhiều vào các giải pháp nhận diện khuôn mặt, nhận diện ký tự - OCR. Những sản phẩm của các giải pháp đó có thể kể đến như: Định danh khách hàng - eKYC, nhận dạng giấy tờ tùy thân, nhận dạng biển số xe. Nhưng công nghệ luôn phát triển không ngừng, và các bài toán đó đã được làm đi làm lại, cải tiến rất nhiều lần và cũng đã dần hoàn thiện, điều đó thúc đẩy mở ra các giải pháp khác cho chúng ta tìm hiểu. Ta nên dự đoán trước xu thế phát triển của tương lai và nghĩ ra giải pháp cho những vấn đề mới.

Hiện tại dù YOLO đã ra tới phiên bản YOLOv5, nhưng phiên bản này đang còn nhiều tranh cãi về tốc độ cũng như độ chính xác, không phải do chính chủ tác giả phát hành. Phiên bản YOLOv4 cũng đã có rất nhiều cải tiến so với YOLOv3 nhưng hiện giờ YOLOv3 vẫn còn rất mạnh về cả tốc độ lẫn độ chính xác, hơn nữa với một người mới tìm hiểu về lĩnh vực này thì tôi chọn phiên bản YOLOv3 để tìm hiểu và triển khai, bởi vì phiên bản càng mới thì kiến trúc của thuật toán càng phức tạp, nếu quá phức tạp sẽ rất khó để tìm hiểu trong thời gian ngắn.

PyTorch là một package được xây dựng dựa trên Python để thay thế Numpy để tận dụng sức mạnh tính toán của GPU và cung cấp tính linh hoạt như một nền tảng phát triển Deep Learning. Pytorch là framework được phát triển bởi Facebook, Inc[[2]](#footnote-2). Đây là một ông lớn về công nghệ đầu tư rất nhiều nguồn lực cho việc phát triển Trí tuệ nhân tạo. Pytorch được phát triển với giấy phép mã nguồn mở do đó nó tạo được cho mình một cộng đồng rất lớn. Một cộng đồng lớn đồng nghĩa với nhiều tài nguyên để học và các vấn đề của ta có thể đã có ai đó giải quyết và chia sẻ với cộng đồng. Pytorch cùng với Tensorflow[[3]](#footnote-3) và Keras[[4]](#footnote-4) là một trong những framework phổ biến được sử dụng trong các bài toán về Deep Learning hiện nay. Đặc biệt, trong các lĩnh vực nghiên cứu, hầu như các tác giả đều sử dụng Pytorch để triển khai bài toán của mình.

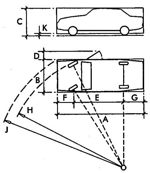
Django là một trong số những web framework bậc cao miễn phí, là mã nguồn mở được tạo ra bởi ngôn ngữ Python dựa trên mô hình mô hình MTV (gồm Model-Template-Views). Hiện framework này được phát triển, quản lý bởi Django Software Foundation. Django ra đời với mục tiêu hỗ trợ thiết kế các website phức tạp dựa trên những CSDL có sẵn. Nó hoạt động dựa theo nguyên lý ‘cắm’ các thành phần và và tái sử dụng để tạo nên các website với ít code, ít khớp nối, có khả năng phát triển và không bị trùng lặp. Lợi thế hàng đầu của Django là khả năng thiết kế, tạo lập website và các ứng dụng nhanh chóng. Vì vậy Django được sử dụng để thực hiện đồ án này.

## **Quá trình nguyên cứu**

Để hiểu rõ hơn về nghiệp vụ liên quan tới bãi đỗ xe, tôi sử dụng các mẫu bãi đỗ xe truyền thống và phổ biến nhất trên internet để tìm hiểu và triển khai giúp ích cho quá trình thu thập hình ảnh và đánh nhãn cũng như nhận diện sau này được dễ dàng hơn.

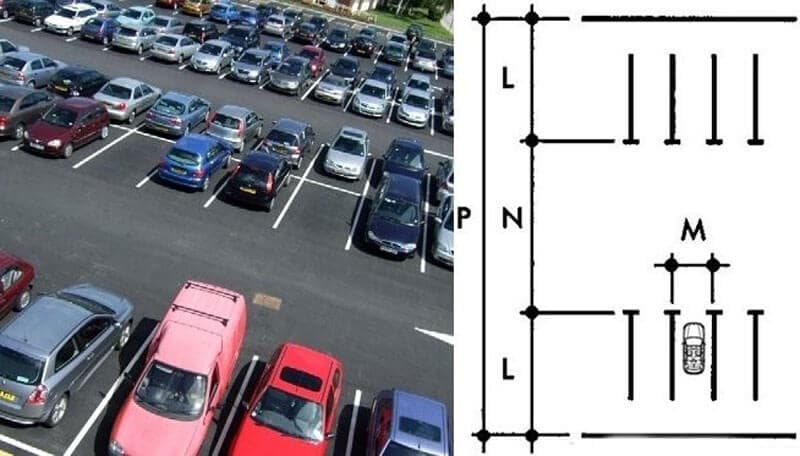
### ***Tiêu chuẩn kích thước của bãi đỗ xe ô tô***

* Kích thước ô tô tiêu chuẩn:



Hình 0.6. Minh họa kích thước ô tô tiêu chuẩn

* Chiều dài xe (A): 4.75 m
* Chiều rộng xe (B): 1.80 m
* Chiều cao xe (C): 1.70 m
* Khoảng mở tối thiểu (D): 0.50 m
* Khoảng cách giữa 2 trục bánh xe (E): 2.90 m
* Khoảng cách từ đầu xe đến trục bánh trước (F): 0.90 m
* Khoảng cách từ đuôi xe đến trục bánh sau (G): 1.10 m
* Đường kính vòng quay xe (H): 13 m
* Khoảng cách từ tâm đường kính quay đến tường (J): 14 m
* Khoảng sáng gầm xe (K): 0.10 m
* Tiêu chuẩn kích thước bãi đỗ xe thông dụng:



Hình 0.7. Tiêu chuẩn kích thước bãi đỗ xe thông dụng

Diễn giải các ký hiệu kích thước chỗ đỗ xe ô tô:

+ L: Chiều dài chỗ đỗ xe ô tô: Từ 4.75m đến 5.5m

+ M: Chiều rộng chỗ đậu xe ô tô; chiều rộng chỗ đỗ xe được quy định như sau:

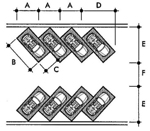
* Chiều rộng chỗ đỗ xe ô tô tiêu chuẩn : Từ 2.4m đến 2.75m
* Chiều rộng chỗ đỗ xe ô tô thời gian ngắn: Từ 2.3m đến 2.6m
* Chiều rộng chỗ đỗ xe ô tô thời gian dài: Từ 2.5m đến 2.75m
* Chiều rộng chỗ đỗ xe ô tô dành cho người khuyết tật: Từ 3m đến 3.5m

+ N: Chiều rộng lối đi ôtô :

* Chiều rộng lối đi ô tô tại làn đường có 1 chiều xe chạy : Từ 6m đến 9.15m
* Chiều rộng lối đi ô tô tại làn đường có 2 chiều xe chạy : Từ 6.95m đến 10.7m

+ P: Chiều rộng tiêu chuẩn bãi đậu xe : Từ 15.5m đến 20.1m

* Tiêu chuẩn kích thước bãi đỗ xe ô tô chéo góc 45 độ:



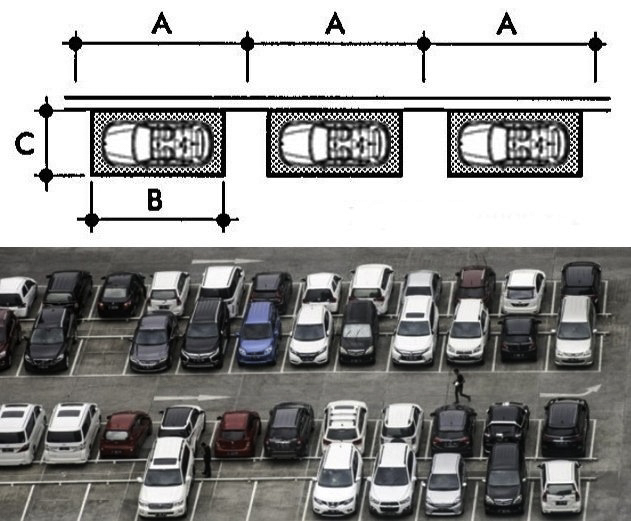
Hình 0.8. Tiêu chuẩn thiết kế bãi đỗ xe 45 độ

+ Tiêu chuẩn:

Bảng 0.1. Tiêu chuẩn kích thước bãi đỗ xe 45 độ (tính theo mét)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Vị trí** | **A** | **B** | **C** | **D** | **E** | **F** |
| Trong nhà | 3.0 | 4.6 | 2.3 | 3.26 | 5.0 | 2.8 |
| Ngoài trời | 3.39 | 5.50 | 2.40 | 3.89 | 5.50 | 2.8 - 3.0 |

* Tiêu chuẩn kích thước bãi đỗ xe ô tô song song:



Hình 0.9. Tiêu chuẩn thiết kế bãi đỗ xe song song

+ Tiêu chuẩn:

Bảng 0.2. Tiêu chuẩn kích thước bãi đỗ xe song song (tính theo mét)

|  |  |  |  |
| --- | --- | --- | --- |
| **Vị trí** | **A** | **B** | **C** |
| Trong nhà | 5.8 | 4.6 | 2.20 - 2.30 |
| Ngoài trời | 6.10 - 6.70 | 5.50 | 2.40 |

### ***Các trạng thái có thể có trong một chỗ đỗ xe ô tô***

Chỉ có 2 trạng thái chính, đó là:

* Trống: Chỗ đó chưa có ô tô nào được đỗ
* Đã đỗ: Chỗ đó đã có ô tô đỗ

### ***Các điều kiện bên ngoài ảnh hưởng đến chất lượng hình ảnh***

* Thời tiết: Mưa, nắng
* Ánh sáng: Nắng đỗ bóng, nắng gắt, nắng nhẹ, râm mát, trời tối bật đèn, sương mù,...
* Các yếu tố gây nhiễu: Cây cối, bụi bặm bám trên camera,…

### ***Một số dạng camera giám sát phổ biến hiện nay***

* **Camera quan sát có dây (Camera Analog): Camera quan sát có dây là loại camera với cảm biến số hóa hình ảnh để xử lí. Camera ghi hình ổn định và chuyên nghiệp cùng sự hỗ trợ của nhiều thiết bị như dây cáp, đầu ghi,... Camera có dây hiện nay đã được cải thiện khá nhiều về cách thức hoạt động và hỗ trợ công nghệ quan sát từ xa qua mạng Internet giúp người dùng dễ dàng quan sát và quản lí thông qua các ứng dụng được cài đặt trên điện thoại thông minh, laptop, máy tính bảng,... có thể quan sát nơi làm việc, nhà ở, bãi đỗ xe,... ở mọi nơi nhờ những tính năng nổi bật của camera có dây. Camera có dây đem lại hình ảnh chất lượng tốt, truyền tải ổn định, tính bảo mật cao.**



Hình 0.10. Minh họa Camera Analog

* **Camera quan sát không dây (Camera Wifi): Camera Wifi có ưu điểm lắp đặt nhanh, sử dụng dễ dàng, thuận tiện vì vậy sản phẩm camera không dây trở nên thông dụng ở thời điểm hiện tại. Camera được kết nối với Wifi ổn định và chỉ cần một thiết bị như smartphone, máy tính bảng, laptop,... có kết nối Internet là có thể giám sát mọi thứ ở bất kỳ đâu. Ngoài ra, một số camera không dây hỗ trợ thẻ nhớ giúp thuận tiện xem lại hình ảnh khi có cần. Camera quan sát không dây tích hợp các tính năng thông minh như cảnh báo chuyển động, đàm thoại 2 chiều, xoay 360 độ,... giúp chống trộm và đảm bảo an ninh an toàn, hiệu quả.**



Hình 0.11. Minh họa Camera Wifi

### ***Quá trình giải quyết bài toán nhận diện vật thể bằng thuật toán YOLOv3***

Có 2 phương án để giải quyết bài toán nhận diện chỗ còn trống và đã đỗ với thuật toán YOLOv3, đó là:

* **Thứ nhất, sử dụng tệp weights được tác giả thuật toán YOLOv3 huấn luyện sẵn với tập dữ liệu COCO (tệp weights và tập dữ liệu COCO sẽ được giới thiệu sau), sau đó nhận diện các vị trí của ô tô, và lấy tổng các vị trí hiện có trong bãi trừ cho số lượng vị trí ô tô đó. Tuy phương án này rút ngắn thời gian huấn luyện mô hình nhưng độ chính xác không cao, dễ bị nhiễu bởi các đối tượng khác bởi vì tập dữ liệu COCO được huấn luyện để nhận dạng lên đến 80 đối tượng, và trong lúc nhận diện có thể có ô tô đang di chuyển sẽ khiến thuật toán nhận diện nhầm. Hơn nữa các vị trí còn trống sẽ không được nhận diện, cho nên thuật toán sẽ không biết khi nào ô tô được đậu đúng vị trí, vì khi ô tô đậu bên ngoài vạch cũng được nhận diện. Cuối cùng phương án này chúng ta cần phải biết trước số lượng tổng các vị trí trong bãi. Đây là phương án tiếp cận quá nhiều bước.**
* **Thứ hai, ta sẽ huấn luyện mô hình để nhận diện được cả 2 trạng thái trống và đã đỗ trong bãi, sau đó tính tổng bằng cách lấy tổng các vật thể ta nhận diện được. Ta sẽ chọn phương án này để triển khai.**

## **Cấu trúc báo cáo đồ án tốt nghiệp**

Cấu trúc đồ án được chia thành các chương như sau:

Tổng quan – Giới thiệu tổng quan về đề tài đồ án tốt nghiệp.

Chương 1: Cơ sở lý thuyết

* Phân biệt các khái niệm liên quan đến Trí Tuệ Nhân Tạo, Thị Giác Máy Tính, Học Máy, Học Sâu, Khai Phá Dữ Liệu.
* Giới thiệu các công nghệ sử dụng.
* Các kiến thức liên quan đến thống kê học sử dụng trong bài làm.
* Tổng quan về xử lý ảnh.
* Tổng quan về Thị Giác Máy Tính và Học Sâu.

Chương 2: Mạng Neural Tích Chập (CNN)

* Giới thiệu Mạng Neural và Mạng Neural lan truyền ngược.
* Giới thiệu Mạng Neural Tích Chập.
* Các thuật toán dựa trên Mạng Neural Tích Chập trong nhận diện đối tượng.
* Giới thiệu về Transfer Learning.
* Giới thiệu các phiên bản của thuật toán YOLO.
* Giới thiệu các chỉ số đánh giá thuật toán.

Chương 3: Thiết kế, triển khai thuật toán và xây dựng website

* Giới thiệu một vài tập dữ liệu phổ biến.
* Thu thập dữ liệu, tiền xử lý, gắn nhãn và huấn luyện thuật toán.
* Triển khai thuật toán YOLOv3 nhận diện qua hình ảnh, video, camera xem trực tiếp màn hình xử lý trên máy tính.
* Kiểm tra thuật toán.
* Xây dựng website giới thiệu sản phẩm và xem màn hình xử lý.

Kết luận và kiến nghị

* Đưa ra kết quả đạt được, những thứ còn tồn tại và hướng phát triển về thuật toán lẫn website trong tương lai.

Tài liệu tham khảo.

# **CƠ SỞ LÝ THUYẾT**

## **1.1 Phân biệt các khái niệm liên quan đến Trí Tuệ Nhân Tạo, Thị Giác Máy Tính, Học Máy, Học Sâu**

***1.1.1 Trí Tuệ Nhân Tạo***

Trí Tuệ Nhân Tạo (Artificial Intelligence – viết tắt là AI) có thể được định nghĩa như một ngành của khoa học máy tính liên quan đến việc tự động hóa các hành vi thông minh. AI là một bộ phận của khoa học máy tính và do đó nó phải được đặt trên những nguyên lý lý thuyết vững chắc, có khả năng ứng dụng được của lĩnh vực này. Có thể hiểu đó là trí tuệ của máy móc được tạo ra bởi con người. Trí tuệ này có thể tư duy, suy nghĩ, học hỏi,... như trí tuệ con người. Xử lý dữ liệu ở mức rộng lớn hơn, quy mô hơn, hệ thống, khoa học và nhanh hơn so với con người.

 Trong lịch sử phát triển AI, các nhà nghiên cứu phân thành 4 hướng tiếp cận chính:

* Hành động như người (acting humanly)
* Suy nghĩ như người (thinking humanly)
* Suy nghĩ hợp lý (thinking rationally)
* Hành động hợp lý (acting rationally)

Một trong những hướng phát triển đó điển hình là bài kiểm tra Turing (Turing test), đây là hướng phát triển AI hành động như người. Bài kiểm tra này được Alan Turing[[5]](#footnote-5) đề xuất vào năm 1950, mục đích để kiểm tra hệ thống máy tính đã đạt đến khả năng thông minh hay chưa. Bài test gồm một người đặt câu hỏi, một người trả lời câu hỏi và một máy tính phản hồi câu hỏi. Nếu trong quá trình trao đổi mà người đặt câu hỏi không thể phân biệt được người hay máy trả lời các câu hỏi này thì máy được xem là thông minh. Đến nay thì hướng tiếp cận này đã đạt được một số thành quả nhất định như:

* **Natural Language Processing**: Máy có khả năng đọc hiểu và giao tiếp bằng ngôn ngữ tự nhiên với người.
* **Knowledge Representation:** Máy có khả năng lưu trữ tri thức thông qua thị giác, thính giác, hay văn bản.
* **Automated Reasoning**: Máy có khả năng sử dụng tri thức đã lưu trữ để trả lời câu hỏi hay đưa ra kết luận hữu ích.
* **Machine Learning:** Máy có khả năng thích nghi với các điều kiện môi trường xung quanh để rút trích ra các nguyên lý từ tri thức thu nhận được phục vụ cho việc ra quyết định.
* **Computer Vision**: Máy có khả năng quan sát và xác định được các đối tượng xung quanh.
* **Robotics**: Máy có khả năng tương tác với đối tượng và di chuyển trong môi trường xung quanh.

Những gì chúng ta đang thực hiện với AI hiện nay nằm trong khái niệm “AI hẹp” (Narrow AI). Công nghệ này có khả năng thực hiện các nhiệm vụ cụ thể một cách tương tự, hoặc tốt hơn con người. Ví dụ về “AI hẹp” trong thực tế như công nghệ phân loại hình ảnh của Pinterest[[6]](#footnote-6) hay nhận diện khuôn mặt để gắn thẻ bạn bè trên Facebook[[7]](#footnote-7).

### ***1.1.2 Học Máy***

Học Máy (Machine Learning – viết tắt là ML) là một thuật ngữ rộng để chỉ hành động dạy máy tính cải thiện một nhiệm vụ mà nó đang thực hiện, là một thành phần trong hướng tiếp cận hành động như người. Cụ thể hơn, ML đề cập tới bất kỳ hệ thống mà hiệu suất của máy tính khi thực hiện một nhiệm vụ sẽ trở nên tốt hơn sau khi hoàn thành nhiệm vụ đó nhiều lần. Hay nói cách khác, khả năng cơ bản nhất của ML là sử dụng thuật toán để phân tích những thông tin có sẵn, học hỏi từ nó rồi đưa ra quyết định hoặc dự đoán về một thứ gì đó có liên quan. Thay vì tạo ra một phần mềm với những hành động, hướng dẫn chi tiết để thực hiện một nhiệm vụ cụ thể, máy tính được “huấn luyện” bằng cách sử dụng lượng dữ liệu và các thuật toán để học cách thực hiện nhiệm vụ.

Các thuật toán ML thường được chia làm 4 nhóm:

* Supervised learning (Học có giảm sát) là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới dựa trên các cặp (input, outcome) đã biết từ trước. Supervised learning là nhóm phổ biến nhất trong các thuật toán Machine Learning. Sử dụng các giải thuật và hướng giải quyết như: Decision tree (cây quyết định), k-NN, naive bayes, SVM, neural network (mạng neural), deep learning (học sâu)…
* Unsupervised Learning (Học không giám sát): Trong thuật toán này, chúng ta không biết được outcome hay nhãn mà chỉ có dữ liệu đầu vào. Thuật toán unsupervised learning sẽ dựa vào cấu trúc của dữ liệu để thực hiện một công việc nào đó, ví dụ như phân nhóm hoặc giảm số chiều của dữ liệu để thuận tiện trong việc lưu trữ và tính toán. Sử dụng các giải thuật và hướng giải quyết như: k-means, hierachical clustering (phân cụm phân cấp).
* Semi-supervised lerning (Học bán giám sát): Các bài toán khi chúng ta có một lượng lớn dữ liệu nhưng chỉ một phần trong chúng được gán nhãn được gọi là Semi-Supervised Learning. Những bài toán thuộc nhóm này nằm giữa hai nhóm được nêu bên trên.
* Reinforcement learning (Học củng cố) là các bài toán giúp cho một hệ thống tự động xác định hành vi dựa trên hoàn cảnh để đạt được lợi ích cao. Hiện tại, Reinforcement Learning chủ yếu được áp dụng vào Lý Thuyết Trò Chơi, các thuật toán cần xác định nước đi tiếp theo để đạt được điểm số cao nhất. Bao gồm các phương pháp: Active (chủ động), passive (thụ động), generalization (tổng quát hóa).

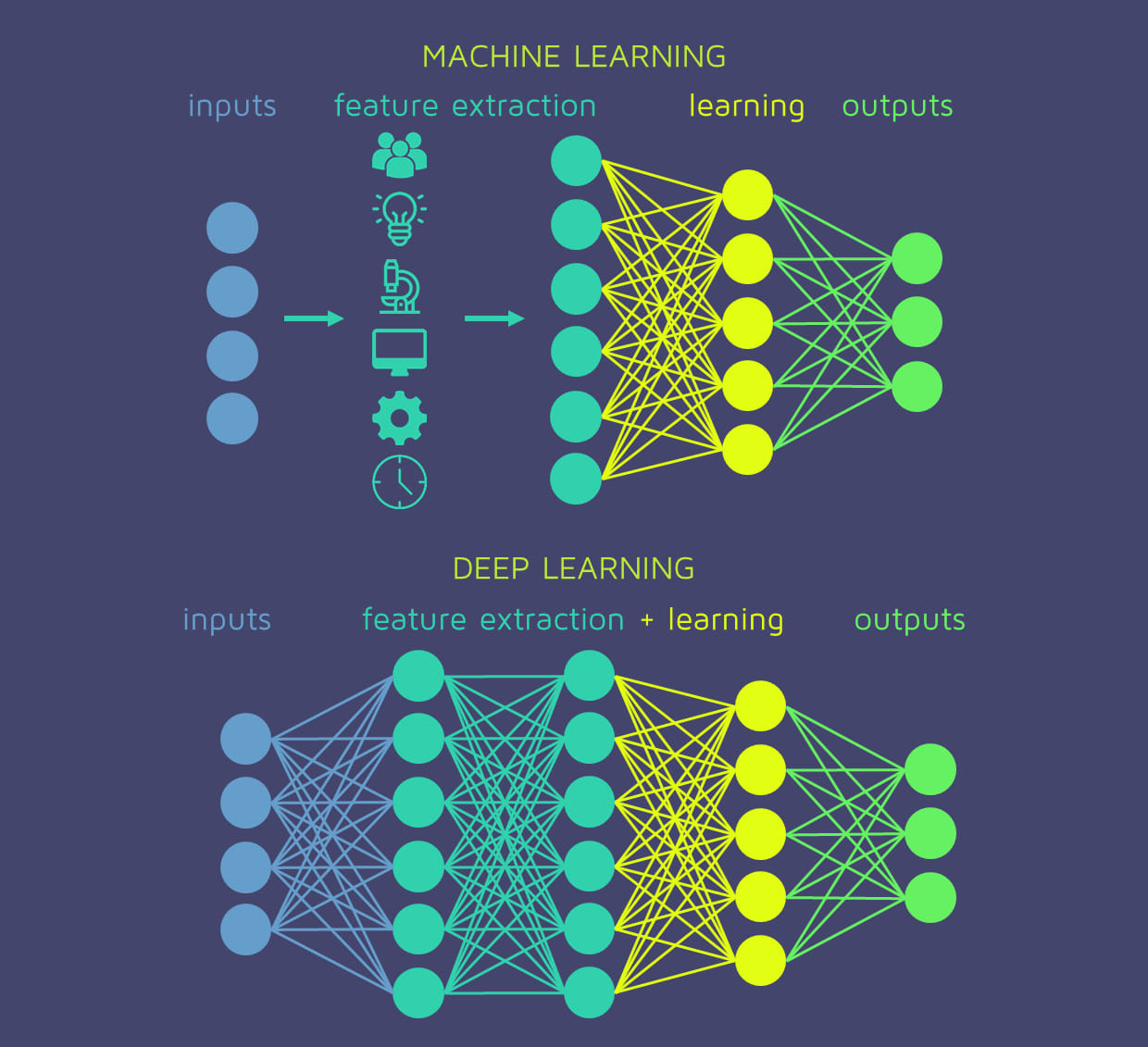
Nếu không có ML, AI hiện tại sẽ bị hạn chế khá nhiều bởi nó mang lại cho máy tính sức mạnh để tìm ra mọi thứ mà không được lập trình rõ ràng. Ví dụ về một loại ML, giả sử ta muốn một chương trình có thể xác định được mèo trong các bức ảnh:

* Đầu tiên, ta cung cấp cho AI một tập hợp các đặc điểm của loài mèo để máy nhận dạng, ví dụ như màu sắc lông, hình dáng cơ thể, kích thước…
* Tiếp theo, ta cung cấp một số hình ảnh cho AI, trong đó một số hoặc tất cả các hình ảnh có thể được dán nhãn "mèo" để máy có thể chọn hiệu quả hơn các chi tiết, đặc điểm có liên quan đến mèo.
* Sau khi máy đã nhận được đủ dữ liệu cần thiết về mèo, nó phải biết cách tìm một con mèo trong một bức tranh - “Nếu trong hình ảnh có chứa các chi tiết X, Y, hoặc Z nào đó, thì 95% khả năng đó là một con mèo”.

***1.1.3 Học Sâu***

Học Sâu (Deep Learning – viết tắt là DL) chỉ là một phương pháp nằm trong hướng giải quyết học có giám sát của ML. Nó như là một loại ML với "neural network" sâu có thể xử lý dữ liệu theo cách tương tự như một bộ não con người có thể thực hiện. Điểm khác biệt chính ở đây là con người không sẽ phải dạy một chương trình Deep Learning biết một con mèo trông như thế nào, mà chỉ cần cung cấp cho nó đủ hình ảnh cần thiết về loài mèo, và nó sẽ tự mình hình dung, tự học. Các bước cần làm như sau:

* Cung cấp cho máy rất nhiều ảnh về mèo.
* Thuật toán sẽ kiểm tra ảnh để xem các đặc điểm, chi tiết chung giữa các bức ảnh.
* Mỗi bức ảnh sẽ được giải mã chi tiết dưới nhiều cấp độ, từ các hình dạng lớn, chung đến các ô nhỏ và nhỏ hơn nữa. Nếu một hình dạng hoặc các đường được lặp lại nhiều lần, thuật toán sẽ gắn nhãn nó như là một đặc tính quan trọng.
* Sau khi phân tích đủ hình ảnh cần thiết, thuật toán giờ đây sẽ biết được các mẫu nào cung cấp bằng chứng rõ ràng nhất về mèo và tất cả những gì con người phải làm chỉ là cung cấp các dữ liệu thô.



Hình 1.1. Sự khác biệt giữa ML và DL

Tóm lại Deep learning là loại ML mà trong đó máy tự đào tạo chính nó. Deep Learning đòi hỏi rất nhiều dữ liệu đầu vào và sức mạnh tính toán hơn là ML, nhưng nó đã bắt đầu được triển khai bởi các công ty công nghệ lớn như Facebook, Amazon[[8]](#footnote-8). Trong đó, một trong những cái tên nổi tiếng nhất về ML là AlphaGo[[9]](#footnote-9), một máy tính có thể chơi cờ vây với chính bản thân nó cho đến khi nó có thể dự đoán những đường đi nước bước chính xác nhất đủ để đánh bại nhiều nhà vô địch trên thế giới.

***1.1.4 Thị Giác Máy Tính***

Thị Giác Máy Tính (Computer Vision – viết tắt là CV) là một lĩnh vực liên ngành của Trí Tuệ Nhân Tạo nhằm mục đích cung cấp cho máy tính và các thiết bị khác có khả năng tính toán hiểu biết ở mức độ cao từ cả hình ảnh và video kỹ thuật số, bao gồm chức năng thu thập, xử lý và phân tích hình ảnh kỹ thuật số. Đây là lý do tại sao Thị Giác Máy Tính, một phần, là một lĩnh vực phụ khác của Trí Tuệ Nhân Tạo, chủ yếu dựa vào Học Máy và các thuật toán Học Sâu để xây dựng các ứng dụng Thị Giác Máy Tính. Ngoài ra, Thị Giác Máy Tính bao gồm một số công nghệ hoạt động cùng nhau: Đồ họa máy tính (Computer graphics), Xử lý hình ảnh (Image processing), Xử lý tín hiệu (Signal processing), Công nghệ cảm biến (Sensor technology), Toán học (Mathematics) hoặc thậm chí Vật lý (Physics).

**Từ các khái niệm trên ta có thể khái quát mối quan hệ giữa AI, ML, DL, CV thông qua sơ đồ sau:**

Graphical user interface

Description automatically generated

Hình 1.2. Sơ đồ mối quan hệ giữa AI, ML, DL, CV

***1.1.5 Sự hỗ trợ của Data Mining***

Data mining, hay được dịch ra là khai phá dữ liệu. Đó là một quá trình khám phá các mẫu trong các tệp dữ liệu rất lớn, liên quan đến các phương thức tại giao điểm của máy học, thống kê học và hệ thống cơ sở dữ liệu. Data mining là một lĩnh vực liên ngành của Khoa học máy tính và thống kê máy tính với mục tiêu tổng thể là trích xuất thông tin (bằng phương pháp thông minh) từ bộ dữ liệu và chuyển đổi thông tin thành cấu trúc dễ hiểu để sử dụng tiếp.

Data mining là bước phân tích của quy trình “khám phá tri thức trong cơ sở dữ liệu”. Ngoài bước phân tích thô, nó còn liên quan đến các khía cạnh quản lý dữ liệu và cơ sở dữ liệu, xử lý trước dữ liệu, cân nhắc mô hình và suy luận, cân nhắc phức tạp, xử lý hậu của các cầu trúc được phát hiện, trực quan hóa và cập nhật trực tuyến.

Trên thực tế, thuật ngữ “Data mining” là một cách gọi sai, vì mục tiêu là khai phá các mẫu và kiến thức từ một lượng lớn dữ liệu, chứ không phải khai phá (mining) dữ liệu. Tuy nhiên, đây là một tên gọi thông dụng và thường được áp dụng cho bất kỳ dạng dữ liệu hoặc xử lý thông tin quy mô lớn nào (thu thập, trích xuất, lưu trữ, phân tích và thống kê) cũng như mọi ứng dụng của hệ thống hỗ trợ quyết định máy tính, bao gồm cả trí tuệ nhân tạo và trí tuệ kinh doanh.

Thống kê học là khoa học nghiên cứu dữ liệu, một ngành học về sự bất định, một dạng phân tích toán học sử dụng các mô hình, sự biểu diễn và tóm tắt định lượng cho một tập hợp dữ liệu thực nghiệm hoặc nghiên cứu thực tế nhất định nhằm phục vụ cho quá trình phân tích, dự đoán và quyết định. Có nhiều công cụ để tìm các thuộc tính có liên quan của dữ liệu nhưng điều này khá gần với toán học thuần túy.

Machine Learning sử dụng các kỹ thuật Data Mining và các thuật toán học tập khác để xây dựng các mô hình về những gì đang xảy ra đằng sau một số dữ liệu để nó có thể dự đoán kết quả trong tương lai. Toán học là cơ sở cho nhiều thuật toán, nhưng điều này hướng tới lập trình nhiều hơn.

Ta có thể nhìn thấy mối quan hệ giữa các lĩnh vực AI, ML, DL, Data Mining trong ngành Khoa học máy tính qua sơ đồ sau:

Graphical user interface

Description automatically generated with medium confidence

Hình 1.3. Sơ đồ mối quan hệ giữa các khái niệm AI, ML, DL và Data Mining

**1.2 Xử lý ảnh**

***1.2.1 Khái niệm***

Xử lý ảnh (Image Processing), viết tắt là XLA, là đối tượng nghiên cứu của lĩnh vực Thị Giác Máy Tính, là quá trình biến đổi từ một ảnh ban đầu sang một ảnh mới với các đặc tính và tuân theo ý muốn của người sử dụng. Xử lý ảnh có thể gồm quá trình phân tích, phân lớp các đối tượng, làm tăng chất lượng, phân đoạn và tách cạnh, gán nhãn cho vùng hay quá trình biên dịch các thông tin hình ảnh của ảnh.

Cũng như xử lý dữ liệu bằng đồ hoạ, xử lý ảnh số là một lĩnh vực của tin học ứng dụng. Xử lý dữ liệu bằng đồ họa đề cập đến những ảnh nhân tạo, các ảnh này được xem xét như là một cấu trúc dữ liệu và được tạo bởi các chương trình. Xử lý ảnh số bao gồm các phương pháp và kỹ thuật biến đổi, để truyền tải hoặc mã hoá các ảnh tự nhiên. Mục đích của xử lý ảnh gồm:

* Biến đổi ảnh làm tăng chất lượng ảnh.
* Tự động nhận dạng ảnh, đoán nhận ảnh, đánh giá các nội dung của ảnh.

Nhận biết và đánh giá các nội dung của ảnh là sự phân tích một hình ảnh thành những phần có ý nghĩa để phân biệt đối tượng này với đối tượng khác, dựa vào đó ta có thể mô tả cấu trúc của hình ảnh ban đầu. Có thể liệt kê một số phương pháp nhận dạng cơ bản như nhận dạng ảnh của các đối tượng trên ảnh, tách cạnh, phân đoạn hình ảnh,… Kỹ thuật này được dùng nhiều trong y học (xử lý tế bào, nhiễm sắc thể), nhận dạng chữ trong văn bản.

***1.2.2 Các quá trình xử lý ảnh***

Diagram

Description automatically generated

Hình 1.4. Các giai đoạn chính trong xử lý ảnh

* Thu nhận ảnh (Image acquisition): Đây là công đoạn đầu tiên mang tính quyết định đối với quá trình XLA. Ảnh đầu vào sẽ được thu nhận qua các thiết bị như camera, sensor, máy scanner,… và sau đó các tín hiệu này sẽ được số hóa. Các thông số quan trọng ở bước này là độ phân giải, chất lượng màu, dung lượng bộ nhớ và tốc độ thu nhận ảnh của các thiết bị.
* Tiền xử lý (image processing):Ở bước này, ảnh sẽ được cải thiện về độ tương phản, khử nhiễu, khử bóng, khử độ lệch,… với mục đích làm cho chất lượng ảnh trở lên tốt hơn nữa, chuẩn bị cho các bước xử lý phức tạp hơn về sau trong quá trình XLA. Quá trình này thường được thực hiện bởi các bộ lọc.
* Phân đoạn ảnh (segmentation):phân đoạn ảnh là bước then chốt trong XLA. Giai đoạn này phân tích ảnh thành những thành phần có cùng tính chất nào đó dựa theo biên hay các vùng liên thông. Tiêu chuẩn để xác định các vùng liên thông có thể là cùng màu, cùng mức xám… Mục đích của phân đoạn ảnh là để có một miêu tả tổng hợp về nhiều phần tử khác nhau cấu tạo lên ảnh thô.
* Trích xuất đặc trưng (feature extraction): Chuyển đổi dữ liệu ảnh thô thành một dạng thích hợp hơn cho việc xử lý. Để chuyển đổi chúng, tùy thuộc vào mục đích mà biểu diễn một vùng ảnh dưới dạng biên hoặc dưới dạng một vùng hoàn chỉnh gồm tất cả những điểm ảnh thuộc về nó. Biểu diễn dạng biên cho một vùng phù hợp với những ứng dụng chỉ quan tâm chủ yếu đến các đặc trưng hình dạng bên ngoài của đối tượng, ví dụ như các góc cạnh và điểm uốn trên biên chẳng hạn. Biểu diễn dạng vùng lại thích hợp cho những ứng dụng khai thác các tính chất bên trong của đối tượng, ví dụ như vân ảnh hoặc cấu trúc xương của nó. Sự chọn lựa cách biểu diễn thích hợp cho một vùng ảnh chỉ mới là một phần trong việc chuyển đổi dữ liệu ảnh thô sang một dạng thích hợp hơn cho các xử lý về sau. Chúng ta còn phải đưa ra một phương pháp mô tả dữ liệu đã được chuyển đổi đó sao cho những tính chất cần quan tâm đến sẽ được làm nổi bật lên, thuận tiện cho việc xử lý chúng.
* Phân tích dữ liệu (data analysis): Đây là bước cuối cùng trong quá trình XLA. Trong đó có quá trình nhận dạng ảnh có thể được nhìn nhận một cách đơn giản là việc gán nhãn cho các đối tượng trong ảnh. Giải thích/phân loại là công đoạn gán nghĩa cho một tập các đối tượng đã được nhận biết.

Chúng ta cũng có thể thấy rằng, không phải bất kỳ một ứng dụng XLA nào cũng bắt buộc phải tuân theo tất cả các bước xử lý đã nêu ở trên, ví dụ như các ứng dụng chỉnh sửa ảnh nghệ thuật chỉ dừng lại ở bước tiền xử lý. Một cách tổng quát thì những chức năng xử lý bao gồm cả nhận dạng và giải thích thường chỉ có mặt trong hệ thống phân tích ảnh tự động hoặc bán tự động, được dùng để rút trích ra những thông tin quan trọng từ ảnh, ví dụ như các ứng dụng nhận dạng ký tự quang học, nhận dạng chữ viết tay… Một điều kiện tiên quyết và quan trọng cho Thị Giác Máy Tính khác với xử lý ảnh là việc thao tác trên nhiều hình ảnh. Trong khi xử lý ảnh hoàn toàn hoạt động với duy nhất một tấm hình ở dạng số hóa, Computer Vision hoạt động phù hợp hơn trên một luồng hình ảnh có mối quan hệ tạm thời đã biết trước.

***1.2.3 Một số hệ màu cơ bản***

**Không gian màu (Colour Space)** được hiểu là các mô hình toán để miêu tả màu sắc. Mỗi không gian màu đều có một tác dụng và ứng dụng trong các bài toán khác nhau.

* **Hệ màu RGB**: là không gian màu phổ biến dùng trong máy tính, máy ảnh, điện thoại và nhiều thiết bị kĩ thuật số khác nhau. Không gian màu này khá gần với cách mắt người tổng hợp màu sắc. Nguyên lý cơ bản là sử dụng 3 màu sắc cơ bản R (red - đỏ), G (green - xanh lục) và B (blue - xanh lam) để biểu diễn tất cả các màu sắc. Thông thường, trong mô hình 24 bit mỗi kênh màu sẽ sử dụng 8bit để biểu diễn, tức là giá trị R, G, B nằm trong khoảng 0 - 255. Bộ 3 số này biểu diễn cho từng điểm ảnh, mỗi số biểu diễn cho cường độ của một màu. Với mô hình màu 24bit thì số màu tối đa có thể tạo ra là 255 x 255 x 255 = 16581375 màu. Một điểm cân lưu ý là với các thư viện đọc ảnh và hiển thị ảnh như matplotlib[[10]](#footnote-10), Pillow[[11]](#footnote-11) thì các ảnh được đọc theo RGB tuy nhiên Opencv[[12]](#footnote-12) đọc ảnh theo các kênh BGR.

Diagram, shape, venn diagram, circle

Description automatically generated

Hình 1.5. Hệ màu RGB

* Không gian màu HSV (còn gọi là HSB) là một cách tự nhiên hơn để mô tả màu sắc, dựa trên 3 số liệu:
* H: (Hue) là phần màu của mô hình màu và được biểu thị dưới dạng một số từ 0 đến 360 độ.
* S: (Saturation) Độ bão hòa là lượng màu xám trong màu, từ 0 đến 100 phần trăm. Một hiệu ứng mờ nhạt có thể có được từ việc giảm độ bão hòa về không để giới thiệu nhiều màu xám hơn.
* B (hay V): (Bright hay Value) Độ sáng hay giá trị hoạt động kết hợp với độ bão hòa và mô tả độ sáng hoặc cường độ của màu sắc, từ 0-100 phần trăm, trong đó 0 là hoàn toàn đen và 100 là sáng nhất và cho thấy màu sắc nhất.

A picture containing text, light

Description automatically generated

Hình 1.6. Hệ màu HSV

### **1.2.4 Biễu diễn ảnh**

Ảnh trong thực tế là một ảnh liên tục cả về không gian và giá trị độ sáng. Để có thể xử lý ảnh bằng máy tính thì cần thiết phải tiến hành số hóa ảnh. Quá trình số hóa biến đổi các tín hiệu liên tục sang tín hiệu rời rạc thông qua quá trình lấy mẫu (rời rạc hóa về không gian) và lượng tử hóa các thành phần giá trị mà về nguyên tắc bằng mắt thường không thể phân biệt được hai điểm liền kề nhau. Các điểm như vậy được gọi là các pixel (Picture Element) hay các phần tử ảnh hoặc điểm ảnh. Ở đây cần phân biệt khái niệm pixel hay đề cập đến trong các hệ thống đồ họa máy tính. Để tránh nhầm lẫn ta gọi khái niệm pixel này là pixel thiết bị. Khái niệm pixel thiết bị có thể xém xét như sau: khi ta quan sát màn hình (trong chế độ đồ họa), màn hình không liên tục mà gồm các điểm nhỏ, gọi là pixel. Mỗi pixel gồm một tập tọa độ (x, y) và màu.

Như vậy mỗi ảnh là tập hợp các điểm ảnh. Khi được số hóa nó thường được biểu diễn bởi mảng 2 chiều I(n,p): n là dòng và p là cột.

Về mặt toán học có thể xem ảnh là một hàm hai biến f(x,y) với x, y là các biến tọa độ. Giá trị số ở điểm (x,y) tương ứng với giá trị xám hoặc độ sáng của ảnh (x là các cột còn y là các hàng). Giá trị của hàm ảnh f(x,y) được hạn chế trong phạm vi của các số nguyên dương.

0 ≤ f(x,y) ≤ fmax.

Với ảnh đen trắng mức xám của ảnh có thể được biểu diễn bởi một số như sau:

*f*



0

)

(

)

(



*d*

*S*

*c*

*k*

*BW*

Trong đó SBW() là đặc tính phổ của cảm biến được sử dụng và k là hệ số tỷ lệ xích. Vì sự cảm nhận độ sáng có tầm quan trọng hàng đầu đối với ảnh đen trắng nên *SBW(**)* được chọn giống như là hiệu suất sáng tương đối. Vì *f* biểu diễn công suất trên đơn vị diện tích, nên nó bao giờ cũng không âm và hữu hạn.

*0≤ f ≤ fmax*

Trong đó *fmax* là giá trị lớn nhất mà *f* đạt được. Trong xử lý ảnh, *f* được chia thang sao cho nó nằm trong một phạm vi thuận lợi nào đó.

Thông thường đối với ảnh xám, giá trị *fmax*là 255 ( 28=256) bởi vì mỗi phần tử ảnh được mã hóa bởi một byte. Khi quan tâm đến ảnh màu ta có thể mô tả màu qua ba hàm số: thành phần màu đỏ qua *R(x,y)*, thành phần màu lục qua *G(x,y)* và thành phần màu lam qua *B(x,y)*. Bộ ba giá trị *R*, *G*, và *B* nhận được từ:



0

)

(

)

(



*d*

*S*

*c*

*k*

*R*

*R*



0

)

(

)

(



*d*

*S*

*c*

*k*

*G*

*G*



0

)

(

)

(



*d*

*S*

*c*

*k*

*B*

*B*

Ở đó *SR(* *), SG(* *) và SB(* *)* theo thứ tự là những đặc tính phổ của các cảm biến (bộ lọc) đỏ, lục và lam. *R*, *G*, *B* cũng không âm và hữu hạn.

Ảnh có thể được biểu diễn theo một trong hai mô hình: mô hình Vector hoặc mô hình Raster.

**Mô hình Vector:** Ngoài mục đích tiết kiệm không gian lưu trữ, dễ dàng hiển thị và in ấn, các ảnh biểu diễn theo mô hình vector còn có ưu điểm cho phép dễ dàng lựa chọn, sao chép, di chuyển, tìm kiếm…Theo những yêu cầu này thì kỹ thuật biểu diễn vector tỏ ra ưu việt hơn. Trong mô hình này, người ta sử dụng hướng vector của các điểm ảnh lân cận để mã hóa và tái tạo lại hình ảnh ban đầu. Các ảnh vector được thu nhận trực tiếp từ các thiết bị số hóa hoặc được chuyển đổi từ các ảnh Raster thông qua các chương trình vector hóa.

**Mô hình Raster:** là mô hình biểu diễn ảnh thông dụng nhất hiện nay. Ảnh được biểu diễn dưới dạng ma trận các điểm ảnh. Tùy theo nhu cầu thực tế mà mỗi điểm ảnh có thể được biểu diễn bởi một hay nhiều bit. Mô hình Raster thuận lợi cho việc thu nhận, hiển thị và in ấn. Các ảnh được sử dụng trong phạm vi của đề tài này cũng là các ảnh được biểu diễn theo mô hình Raster.

Khi xử lý các ảnh Raster chúng ta có thể quan tâm đến mối quan hệ trong vùng lân cận của các điểm ảnh. Các điểm ảnh có thể xếp hàng trên một lưới (raster) hình vuông, lưới hình lục giác hoặc theo một cách hoàn toàn ngẫu nhiên với nhau.

Diagram

Description automatically generated with medium confidence

Hình 1.7. Quan hệ trong vùng lân cận giữa các điểm ảnh

Cách sắp xếp theo hình vuông là được quan tâm đến nhiều nhất và có hai loại:

điểm 4 láng giềng (4 liền kề) hoặc 8 láng giềng (8 liền kề). Với điểm 4 láng giềng, một điểm ảnh I(i, j) sẽ có điểm kế cận theo 2 hướng i và j; trong khi đó với điểm 8 láng giềng, điểm ảnh I(i, j) sẽ có 4 điểm kế cận theo 2 hướng i, j và 4 điểm kế cận theo hướng chéo 45 o (Xem hình 1.7).

***1.2.5 Độ phân giải ảnh***

Độ phân giải ( Resolution) của ảnh là mật độ điểm ảnh được ấn định trên một ảnh số được hiển thị. Khoảng cách giữa các điểm ảnh phải được chọn sao cho mắt người vẫn thấy được sự liên tục của ảnh. Việc lựa chọn khoảng cách thích hợp tạo nên một mật độ phân giải và được phân bố theo trục x và y trong không gian hai chiều.

***1.2.6 Phạm vi ứng dụng của xử lý ảnh***

Xử lý ảnh đã đem lại nhiều ứng dụng trong nhiều lĩnh vực khác nhau: y học, khoa học hình hình sự, khí tượng thuỷ văn, quản lý, sản xuất...

**1.3** **Một số kiến thức về thống kê học sử dụng trong bài làm**

***1.3.1 Biến đổi dữ liệu***

Biến đổi dữ liệu (data transformation) là ứng dụng các hàm toán học xác định cho từng điểm trong tập dữ liệu, nghĩa là, mỗi điểm dữ liệu zᵢ được thay thế bằng giá trị biến đổi yᵢ = f, trong đó f là một hàm.

**Phân phối chuẩn (normal distribution):**

Là dạng phân phối xác suất quan trọng nhất trong thống kê.

Một biến có dạng phân phối chuẩn khi phân phối xác suất của nó có dạng đường cong hình chuông “bell-shaped curve”. Mean nằm chính giữa và các giá trị của biến xoay quanh mean. Trong đó các giá trị mean (giá trị trung bình), median (trung vị), mode (yếu vị) có giá trị như nhau và được xác định bởi 2 tham số: Giá trị trung bình (mean) và phương sai (variance).

Tuy vậy không phải lúc nào dữ liệu trong thực tế cũng theo phân phối chuẩn, chúng thường bị lệch và làm cho kết quả phân tích không hợp lệ, để giải quyết vấn đề này ta sử dụng phép biến đổi logarit.

Diagram

Description automatically generated

Hình 1.8. Minh họa phân phối chuẩn

**Logarithm transformation (Phép biến đổi logarit)**:

Phép biến đổi logarit là một phương pháp biến đổi dữ liệu, trong đó nó thay thế mỗi biến x bằng một Log(x). Việc lựa chọn cơ số logarit thường do nhà phân tích phụ thuộc vào mục đích của mô hình thống kê. Trong bài viết này, ta sẽ tập trung vào sự biến đổi logarit tự nhiên (cơ số e), ký hiệu là ln.

Khi dữ liệu liên tục ban đầu không tuân theo đường cong hình chuông, ta có thể dùng phép biến đổi logarit lên dữ liệu này để làm cho dữ liệu này “chuẩn” nhất có thể để kết quả phân tích thống kê từ dữ liệu này trở nên hợp lệ hơn. Nói cách khác, việc chuyển đổi logarit làm giảm hoặc loại bỏ độ lệch của dữ liệu ban đầu. Lưu ý quan trọng ở đây là dữ liệu gốc phải tuân theo hoặc gần đúng theo phân phối log-normal. Nếu không, phép biến đổi logarit sẽ không hoạt động.

Chart, histogram

Description automatically generated

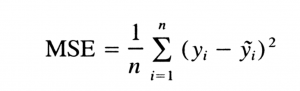
Hình 1.9. Minh họa tác động của phép biến đổi logarit

### ***1.3.2 Sai số toàn phương trung bình (Mean squared error):***

Mean squared error, viết tắt là MSE, là trung bình của bình phương các sai số, tức là sự khác biệt giữa các ước lượng và những gì được đánh giá. MSE là một hàm rủi ro, tương ứng với giá trị kỳ vọng của sự mất mát sai số bình phương hoặc mất mát bậc hai. Sự khác biệt xảy ra do ngẫu nhiên, hoặc vì các ước lượng không tính đến thông tin có thể cho ra một ước tính chính xác hơn.

MSE đánh giá chất lượng của một ước lượng (ví dụ, một hàm toán học lập bản đồ mẫu dữ liệu của một tham số của dân số từ đó các dữ liệu được lấy mẫu) hoặc một yếu tố dự báo (ví dụ, một bản đồ chức năng có số liệu vào tùy ý để một mẫu của các giá trị của một số biến ngẫu nhiên). Định nghĩa của một MSE khác với những gì tương ứng cho dù là một trong những mô tả một ước lượng, hay một yếu tố dự báo.

**Công thức tính MSE:**



Với: yi  là biến độc lập, ỹi là giá trị ước lượng, n là số các giá trị dự báo

**1.4 Phần mềm Nicepage**

Nicepage là một trình xây dựng trang web mới mẻ, khác biệt với các trình xây dựng phổ biến khác trên thị trường. Nó nhằm mục đích trả lại quyền tự do sáng tạo cho các nhà thiết kế web, với trình chỉnh sửa kéo và thả nâng cao cho phép nhà thiết kế web tự do sáng tạo tối đa.

**Các tính năng chính:**

* Tự do kéo thả các thành phần của trang web
* Hỗ trợ Responsive
* Hỗ trợ các hiệu ứng animation
* Hỗ trợ các khối tạo sẵn
* Hỗ trợ nhiều mẫu trang web tạo sẵn

Cung cấp miễn phí giới hạn tính năng và bản trả phí sử dụng nhiều tính năng hơn. Tuy nhiên với nhu cầu thiết kế trang web đơn giản với thời gian nhanh chóng và miễn phí nên tôi chọn phần mềm này để thiết kế trang web của mình.

**1.5 Ngôn ngữ lập trình python**

***1.5.1 Giới thiệu về Python***

Python là ngôn ngữ lập trình hướng đối tượng, cấp cao, thông dịch, mạnh mẽ, được tạo ra bởi Guido van Rossum. Nó dễ dàng để tìm hiểu và đang nổi lên như một trong những ngôn ngữ lập trình nhập môn tốt nhất cho người lần đầu tiếp xúc với ngôn ngữ lập trình. Python hoàn toàn tạo kiểu động và sử dụng cơ chế cấp phát bộ nhớ tự động. Python có cấu trúc dữ liệu cấp cao mạnh mẽ và cách tiếp cận đơn giản nhưng hiệu quả đối với lập trình hướng đối tượng. Cú pháp lệnh của Python là điểm cộng vô cùng lớn vì sự rõ ràng, dễ hiểu và cách gõ linh động làm cho nó nhanh chóng trở thành một ngôn ngữ lý tưởng để viết script và phát triển ứng dụng trong nhiều lĩnh vực, ở hầu hết các nền tảng.

***1.5.2 Lý do Python được sử dụng cho các dự án về AI và Machine Learing***

* **Python đơn giản và nhất quán:** Python cho phép các lập trình viên viết code ngắn gọn và dễ đọc, các lập trình viên sẽ tập trung được thời gian, trí lực vào giải quyết bài toán của Học Máy thay vì gặp rắc rối với kỹ thuật của ngôn ngữ. Và vì Python là ngôn ngữ có mục đích chung, nó có thể thực hiện một tập hợp các tác vụ Học Máy phức tạp và cho phép xây dựng các nguyên mẫu nhanh chóng, kiểm thử sản phẩm của mình cho mục đích học máy tốt hơn.
* **Python có nhiều lựa chọn về thư viện và framework:** Việc thực hiện các thuật toán AI và ML có thể rất khó và mất nhiều thời gian nên có một môi trường được cấu trúc tốt và được thử nghiệm tốt là rất quan trọng, Vì thế, để giảm thời gian phát triển dự án, các lập trình viên chuyển sang một số framework và thư viện của Python. Một framework / thư viện có thể được hiểu và các code được viết sẵn mà các lập trình viên có thể sử dụng ngay để giải quyết các tác vụ lập trình phổ biến. Quan điểm là "không phát minh lại bánh xe". Python có một kho công nghệ phong phú bao gồm rất nhiều thư viện cho trí tuệ nhân tạo và học máy: numpy, opencv, keras, TensorFlow,…
* **Python độc lập với nền tảng:** Độc lập với nền tảng tức là một ngôn ngữ lập trình hoặc framework cho phép các lập trình viên triển khai mọi thứ trên một máy và sử dụng chúng trên một máy khác mà không có bất kỳ thay đổi nào (hoặc chỉ thay đổi tối thiểu).
* **Cộng đồng Python rất lớn:** Trong Khảo sát Lập trình viên năm 2020 của StackOverflow[[13]](#footnote-13), Python đứng thứ 4 trong top 10 ngôn ngữ lập trình phổ biến nhất, điều này có nghĩa là ta có thể tìm thấy các tài liệu học Python, cộng đồng hỗ trợ dễ dàng. Qua nhiều năm phát triển (từ lúc Python được sinh ra năm 1994) cộng đồng Python AI đã phát triển trên toàn thế giới, kể cả Việt Nam. Có nhiều diễn đàn, group tích cực trao đổi về kinh nghiệm và giải pháp liên quan đến Học máy. Đối với bất kỳ nhiệm vụ nào ta cần giải quyết, tỷ lệ khá cao là đã có người nào đó trên cộng đồng đã xử lý thành công.

***1.5.3 Một số thư viện liên quan***

- Numpy là một thư viện lõi phục vụ cho khoa học máy tính của Python, hỗ trợ cho việc tính toán các mảng nhiều chiều, có kích thước lớn với các hàm đã được tối ưu áp dụng lên các mảng nhiều chiều đó. Numpy đặc biệt hữu ích khi thực hiện các hàm liên quan tới Đại Số Tuyến Tính. Trong numpy, chiều của mảng gọi là axes, trong khi số chiều gọi là rank.

- Pandas trong python là một thư viện mã nguồn mở, hỗ trợ đắc lực trong thao tác dữ liệu. Đây cũng là bộ công cụ phân tích và xử lý dữ liệu mạnh mẽ của ngôn ngữ lập trình python. Thư viện này được sử dụng rộng rãi trong cả nghiên cứu lẫn phát triển các ứng dụng về khoa học dữ liệu. Thư viện này sử dụng một cấu trúc dữ liệu riêng là Dataframe. Pandas cung cấp rất nhiều chức năng xử lý và làm việc trên cấu trúc dữ liệu này. Chính sự linh hoạt và hiệu quả đã khiến cho pandas được sử dụng rộng rãi.

**1.6 Django framework**

### ***1.6.1 Ưu điểm của Django:***

1. Sử dụng ngôn ngữ lập trình Python
2. Django đang là giải pháp cốt lỗi trong các lĩnh vực: IOT, các Công ty Blue chip, IT giants (NASA, Google,…), các công ty tài chính
3. Sử dụng ORM, Mô hình MVC (MTV)
4. Hỗ trợ Multi-Site, Đa ngôn ngữ
5. Nhiều tài liệu và cộng đồng hỗ trợ
6. Hỗ trợ giao diện quản lý Admin
7. Dễ dàng mở rộng

### ***1.6.2 Mô hình MTV (Model-Template-Views) trong Django***

Django sử dụng mô hình MTV tương tự như mô hình MVC (Model-View-Controller) trong các framework khác.

Diagram

Description automatically generated

Hình 1.10. Mô hình MTV của Django

Bảng 1.1. Mô tả chức năng các thành phần trong mô hình MTV.

| **Thành phần** | **Mô tả** |
| --- | --- |
| **Model** | Nơi thiết kế ra những table cho database, từ đó Django ORM đã cung cấp những phương thức xử lý, nghiệp vụ lên database |
| **Template** | Là những mẫu được thiết kế ra và xử lý output ra mã html/css cho trang web |
| **Views** | Các hàm để xử lý khi có request từ người dùng |

### ***1.6.3 Cấu trúc Project Django***

* **\_\_init\_\_.py**: Đây là 1 file cơ bản trong Python dùng để biến folder chứa nó thành package, giúp import các file
* **setttings.py**: Đây là file cấu hình project. (VD: cấu hình database, đặt múi giờ, cài thêm thư viện, ...)
* **urls.py**: Đây là file giúp tạo các đường dẫn urls của trang web để liên kết các webpage lại với nhau
* **wsgi.py**: Đây là file giúp chúng ta deploy project lên server

**1.7 Pytorch Framework**

### ***Tensor trong Pytorch***

Tensor cũng giống như numpy array nhưng được chuyển sang tensor để sử dụng tính toán trên GPU. Các phép toán, biến đổi và các hoạt động cơ bản của numpy đều có thể thực hiện được trên Tensor. Việc chuyển đổi từ numpy array sang tensor và ngược lại được Pytorch hỗ trợ rất linh hoạt bằng lệnh *from\_numpy()* và *numpy()*.

### ***1.7.2 CUDA tensor***

**a) Bộ Xử Lý Đồ Họa (GPU)**

GPU là một bộ xử lý có khả năng xử lý tốt các phép tính chuyên biệt. Điều này trái ngược với bộ xử lý trung tâm (CPU), là bộ xử lý có khả năng xử lý tốt các tính toán chung. CPU là bộ xử lý cung cấp năng lượng cho hầu hết các tính toán điển hình trên các thiết bị điện tử.

GPU có thể nhanh hơn nhiều so với CPU, Tuy nhiên, không phải luôn luôn là như vậy. Tốc độ của GPU so với CPU phụ thuộc vào loại tính toán được thực hiện. Loại tính toán phù hợp nhất cho GPU là loại tính toán có thể được thực hiện song song.

Về mặt kiến ​​trúc, sự khác biệt chính giữa CPU và GPU là CPU thường có số lõi hạn chế để thực hiện các phép toán số học. Ngược lại, một GPU có thể có hàng trăm và hàng nghìn lõi.

Chart

Description automatically generated

Hình 1.11. So sánh kiến trúc CPU và GPU

**b) Tính toán song song**

Tính toán song song là một loại tính toán mà bằng một phép tính cụ thể được chia thành các phép tính nhỏ hơn độc lập có thể được thực hiện đồng thời. Các phép tính kết quả sau đó được tổng hợp lại hoặc đồng bộ hóa để tạo thành kết quả của phép tính lớn ban đầu. Số lượng tác vụ mà một tác vụ lớn hơn có thể được chia thành tùy thuộc vào số lượng lõi chứa trên một phần cứng cụ thể. Lõi là các đơn vị thực sự thực hiện tính toán trong một bộ xử lý nhất định và CPU thường có bốn, tám hoặc mười sáu lõi trong khi GPU có thể có hàng nghìn.

Như vậy chúng ta có thể kết luận rằng tính toán song song được thực hiện bằng GPU và các tác vụ phù hợp nhất để giải quyết bằng GPU là các tác vụ có thể được thực hiện song song. Nếu tính toán có thể được thực hiện song song, chúng ta có thể tăng tốc tính toán của mình bằng cách sử dụng các phương pháp lập trình song song và GPU.

Tuy nhiên việc chuyển các tác vụ tính toán tương đối nhỏ sang GPU sẽ không làm tăng tốc và thậm chí có thể làm chậm hơn. GPU hoạt động tốt cho các tác vụ có thể được chia thành nhiều tác vụ nhỏ hơn và nếu một tác vụ tính toán đã nhỏ, chúng ta sẽ không thu được nhiều lợi ích bằng cách chuyển tác vụ đó sang GPU. Vì lý do này, việc chỉ sử dụng CPU khi mới bắt đầu thường được ưu tiên hơn và khi chúng ta giải quyết các vấn đề phức tạp hơn thì lúc đó hãy nghĩ đến việc sử dụng GPU nhiều hơn.

**c) CUDA**

CUDA là nền tảng lập trình tính toán và mô hình lập trình được phát triển bởi Nvidia[[14]](#footnote-14) cho việc tính toán nói chung trên chính các nhân GPU của hãng. CUDA cho phép lập trình viên tăng tốc các ứng dụng tính toán bằng cách khai thác sức mạnh của GPU cho các phần tính toán song song.

**d) CUDA trong Deep Learning**

Deep learning có yêu cầu tính toán tốc độ rất cao. Ví dụ, để train cho model về dịch thuật Google Translate năm 2016, Google Brain và Google Translate đã thử nghiệm hàng trăm lần, mỗi lần kéo dài một tuần để chạy TensorFlow sử dụng GPU. Họ đã mua 2000 server được trang bị GPU từ Nvidia. Nếu không có GPU, việc training sẽ mất hàng tháng thay vì trong 1 tuần training.

Hầu hết các Deep Learning framework sử dụng thư viện cuDNN cho việc tính toán Deep Neural Network. Thư viện này cũng rất quan trọng trong việc training của các mô hình Deep Learning. Khi CUDA và cuDNN được cải tiến liên tục qua các phiên bản, các ứng dụng Deep Learning sẽ tiếp tục được hưởng lợi về mặt tốc độ.

**e) CUDA tensor trong PyTorch**

Thư viện CUDA của PyTorch cho phép theo dõi GPU đang sử dụng và khiến bất kỳ bộ tensor nào được tạo được tự động gán cho thiết bị đó, để đưa một tensor lên một thiết bị bất kì ta sử dụng hàm *to(), cuda().* Sau khi một tensor được cấp phát, ta có thể thực hiện các thao tác với nó và kết quả cũng được gán cho cùng một thiết bị. Kiểm tra thiết bị có hỗ trợ CUDA không bằng lệnh *torch.cuda.is\_available ().*

* **Thực thi không đồng bộ:**

Các hoạt động của GPU là không đồng bộ theo mặc định để cho phép thực hiện song song số lượng lớn hơn các phép tính. Các hoạt động không đồng bộ nói chung là vô hình đối với người dùng vì PyTorch tự động đồng bộ hóa dữ liệu được sao chép giữa CPU và GPU hoặc GPU và GPU. Ngoài ra, các hoạt động được thực hiện theo thứ tự xếp hàng. Điều này đảm bảo rằng các hoạt động được thực thi theo cùng một kiểu như thể các phép tính là đồng bộ.

Một điểm cần lưu ý về việc sử dụng các hoạt động không đồng bộ hay đồng bộ là với các phép đo thời gian. Với các hoạt động không đồng bộ, các phép đo sẽ không chính xác. Để giải quyết vấn đề này trong khi vẫn bật chế độ không đồng bộ có thể gọi *torch.cuda.synchronize ()* trước khi đo hoặc có thể sử dụng *torch.cuda.Event* để ghi lại thời gian.

* **PyTorch Autograd:**

**Chức năng của autograd:** Tự động tính toán đạo hàm trên toàn bộ các toán tử của tensor. Nó là một framework được định nghĩa trong quá trình chạy, có nghĩa rằng quá trình lan truyền ngược được xác định khi mà code được chạy, và do đó mỗi vòng lặp có thể có kết quả thay đổi tham số theo lan truyền ngược khác nhau.

**Theo dõi lịch sử của tensor torch:** torch.tensor là package khởi tạo các tensor torch. Mỗi một tensor torch sẽ có 1 thuộc tính là *.requires\_grad*, nếu ta set thuộc tính này về *True*, các toán tử triển khai trên tensor sẽ được theo dõi. Khi kết thúc quá trình lan truyền thuận (hoặc quá trình tính toán output) ta có thể gọi *.backward()* và mọi tính toán gradient sẽ được tự động thực hiện dựa trên lịch sử đã được lưu lại. Các gradient cho tensor này sẽ được tích lũy và xem tại thuộc tính *.grad*.

Ngoài ra để ngăn tensor lưu lại lịch sử (và sử dụng bộ nhớ), chúng ta cũng có thể bao quanh code block triển khai tensor với hàm *with torch.no\_grad():* nó rất hữu ích trong trường hợp đánh giá model khi sử dụng lệnh *model.eval()* bởi vì khi thuộc tính *requires\_grad = True* thì model sẽ có thể được cập nhật tham số. Nhưng quá trình đánh giá model sẽ không cần cập nhật tham số nên chúng ta không cần áp dụng gradient lên chúng. Đơn giản là set *requires\_grad = False*.

***1.7.3 Cấu trúc của một mô hình Học Sâu trong PyTorch***

Chúng ta có thể tạo một mô hình là một lớp bằng cách kế thừa **nn.Module**, điều này cho phép tạo một mô hình học sâu dưới dạng một lớp. Lớp này có 2 hàm bắt buộc phải có đó là **init** và **forward**:

* **init** là một hàm khởi tạo nhận vào các biến, tham số giúp ta có thể khởi tạo các biến, hàm của đối tượng được khởi tạo. Vì Class này kế thừa **nn.Module** nên khi khởi tạo một đối tượng mới của Class thì phải khởi tạo lớp kế thừa nên trong hàm init luôn phải có *super().\_\_init\_\_()* . Ngoài ra hàm khởi tạo này chúng ta có thể khởi tạo các layer custom sử dụng trong mô hình, tạo backbone bằng cách load mô hình pretrained, khởi tạo và thực thi một số hàm khác.
* **forward** là hàm nhận vào là dữ liệu input ban đầu. Dữ liệu sẽ đi lần lượt qua từng layer của model và trả về output của model.

**1.8 Giới thiệu OpenCV**

***1.8.1 Khái niệm***

OpenCV viết tắt cho Open Source Computer Vision Library. OpenCV là thư viện nguồn mở hàng đầu cho Computer Vision và Machine Learning, và hiện có thêm tính năng tăng tốc GPU cho các hoạt động theo real-time.

OpenCV có một cộng đồng người dùng khá hùng hậu hoạt động trên khắp thế giới bởi nhu cầu cần đến nó ngày càng tăng theo xu hướng chạy đua về sử dụng computer vision của các công ty công nghệ.

***1.8.2 Ứng dụng của OpenCV***

* Hình ảnh street view
* Kiểm tra và giám sát tự động
* Robot và xe hơi tự lái
* Phân tích hình ảnh y học
* Tìm kiếm và phục hồi hình ảnh/video
* Phim – cấu trúc 3D từ chuyển động
* Nghệ thuật sắp đặt tương tác

### ***1.8.3 Các tính năng chính***

* Xử lý và hiển thị Hình ảnh/ Video/ I/O (core, imgproc, highgui)
* Phát hiện các vật thể (objdetect, features2d, nonfree)
* Geometry-based monocular hoặc stereo computer vision (calib3d, stitching, videostab)
* Computational photography (photo, video, superres)
* Machine learning & clustering (ml, flann)
* CUDA acceleration (gpu)

### ***1.8.4 Cấu trúc dữ liệu ảnh trong OpenCV***

Cấu trúc dữ liệu hình ảnh trong OpenCV được tổ chức dưới dạng một ma trận (số học) 3 chiều. Thứ tự các chiều của ma trận được sắp xếp theo chiều cao, chiều rộng và kênh màu (height, width, channel). Mỗi phần tử trong ma trận này có kiểu dữ liệu là số nguyên (0-255) hoặc số thực (0-1) mô tả giá trị của mức sáng (intensity).

Trong Xử Lý Ảnh, mức sáng của ảnh chạy trong miền giá trị 0-255. Màu đen mang giá trị 0, màu trắng mang giá trị 255. Màu xám đậm sẽ có giá trị lệch về 0 nhiều, và màu xám nhạt lệch về phía 255. Do đó, đối với ảnh xám ta chỉ cần một kênh màu để mô tả và lưu trữ, đối với ảnh màu được biểu diễn theo hệ màu BGR được giới thiệu ở trên, và ta cần tới 3 kênh màu để lưu trữ.

OpenCV (phiên bản Python) sử dụng cấu trúc dữ liệu Numpy, do đó ảnh đọc lên bằng câu lệnh *cv2.imread()* sẽ trả ra một numpy array. Các thao tác ta thực hiện sẽ xử lý trên numpy array. Do đó, ta thoải mái sử dụng các hàm có trong thư viện numpy lên ma trận ảnh.

## **1.9 Darknet: Open Source Neural Networks**

Darknet là một framework open source chuyên biệt về nhận diện đối tượng được viết bằng ngôn ngữ C và CUDA. Các mô hình được huấn luyện trên Darknet nhanh hơn, đồng thời Darknet dễ cài đặt và hỗ trợ tính toán CPU và GPU. Cộng đồng sử dụng Darknet đông đảo, được hỗ trợ nhiệt tình. Để không tốn thời gian xây dựng lại mô-đun huấn luyện thuật toán YOLOv3 thì tôi sử dụng thư viện này để tận dụng những mã nguồn có sẵn để thực hiện huấn luyện mô hình song song với quá trình triển khai thuật toán, và tinh chỉnh các thông số trong lúc huấn luyện.

## **1.10 Google Colab**

Google Colaboratory (gọi tắt là Google Colab hay Colab) là một sản phẩm của Google Research. Colab dựa trên Jupyter Notebook, người dùng có thể viết và thực thi đoạn mã python thông qua trình duyệt và đặc biệt rất phù hợp với Data Analysis, Machine Learning và giáo dục.

Sự phát triển mạnh mẽ của Machine learning và Deep learning trong những năm gần đây không chỉ bởi các thuật toán, các mô hình tân tiến liên tiếp ra đời mà còn bởi sự phát triển không ngừng của phần cứng, đặc biệt là GPU. Việc tính toán về toán học cho Deep Learning trên CPU có thể mất hàng tháng! Nhưng những tính toán này có thể được gán cho GPU để thực hiện nhanh hơn. Việc huấn luyện Neural Network trên CPU không được khuyến nghị. GPU cần thiết cho việc tính toán ở mức độ cao.

Như chúng ta đều biết, GPU rất đắt tiền và các nền tảng đám mây như AWS[[15]](#footnote-15) và GCP[[16]](#footnote-16) cũng vậy. Ta có thể giải quyết vấn đề này bằng Colab, nó cung cấp một loạt các GPU miễn phí như GPU Nvidia Tesla K80, Tesla T4, Tesla P4, tùy thuộc vào độ may rủi mỗi phiên (session) mà Google xếp ta vào, mỗi loại GPU có giá lên đến vài chục triệu đồng và là có sức mạnh to lớn về hiệu suất. Vì vậy Google Colab được tôi sử dụng để huấn luyện mô hình trong đồ án này.

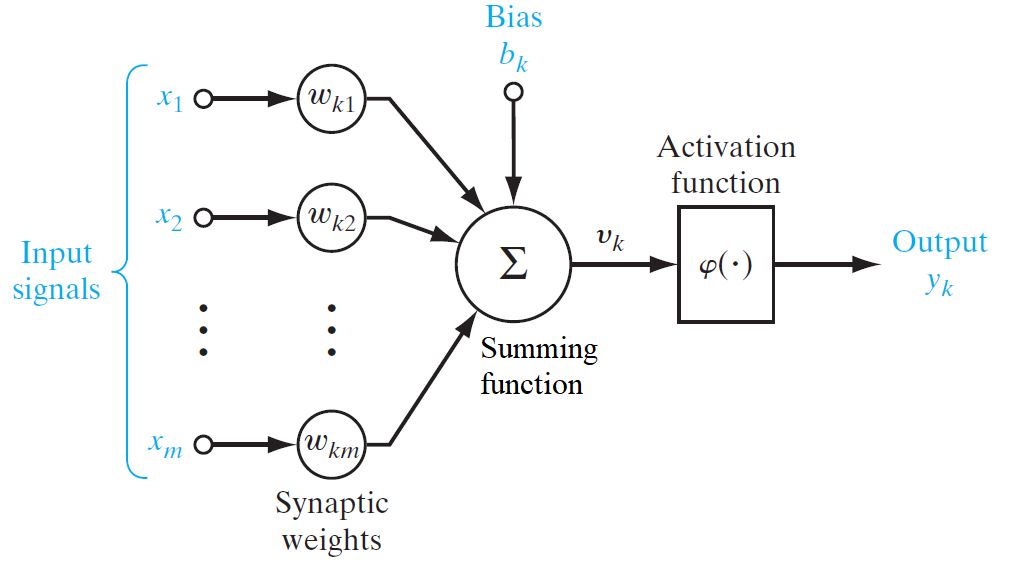
1. **MẠNG NEURAL TÍCH CHẬP**

**2.1 Giới thiệu về Mạng Neural và Mạng Neural lan truyền ngược**

***2.1.1 Định nghĩa***

Mạng nơron nhân tạo, Artificial Neural Network (ANN) là một mô hình xử lý thông tin phỏng theo cách thức xử lý thông tin của các hệ nơron sinh học. Nó được tạo nên từ một số lượng lớn các phần tử (nơron) kết nối với nhau thông qua các liên kết (trọng số liên kết) làm việc như một thể thống nhất để giải quyết một vấn đề cụ thể nào đó. Một mạng nơron nhân tạo được cấu hình cho một ứng dụng cụ thể (nhận dạng mẫu, phân loại dữ liệu,...) thông qua một quá trình học từ tập các mẫu huấn luyện. Về bản chất học chính là quá trình hiệu chỉnh trọng số liên kết giữa các nơron. Các neuron này được gọi là các Perceptron.

***2.1.2 Cấu tạo Mạng Neural***



Hình 1.12. Cấu trúc của một Neural nhân tạo được gán nhãn k

Các thành phần cơ bản của một nơron nhân tạo bao gồm:

• Tập các đầu vào: Là các tín hiệu vào (input signals) của nơron, các tín hiệu này thường được đưa vào dưới dạng một vector N chiều.

• Tập các liên kết: Mỗi liên kết được thể hiện bởi một trọng số liên kết – Synaptic weight. Trọng số liên kết giữa tín hiệu vào thứ j với nơron k thường được kí hiệu là wkj. Thông thường, các trọng số này được khởi tạo một cách ngẫu nhiên ở thời điểm khởi tạo mạng và được cập nhật liên tục trong quá trình học mạng.

• Hàm tổng (Summing function): Thường dùng để tính tổng của tích các đầu vào với trọng số liên kết của nó.

• Ngưỡng (còn gọi là một độ lệch - bias): Ngưỡng này thường được đưa vào như một thành phần của hàm truyền.

• Một hàm kích hoạt (activation function) dùng để đưa các tín hiệu đầu ra của nơron vào một miền giá trị nhất định hoặc vào một tập hợp các giá trị cố định.

• Đầu ra: Là tín hiệu đầu ra của một nơron, với mỗi nơron sẽ có tối đa là một đầu ra.

Diagram, text

Description automatically generated with medium confidenceXét về mặt toán học, cấu trúc của một nơron nhãn k, được mô tả bằng cặp biểu thức sau:

Trong đó: x1, x2, ..., xm: là các tín hiệu vào; (wk1, wk2,..., wkm) là các trọng số liên kết của nơron thứ k; uk là hàm tổng; bk là độ lệch; *φ*(.) là hàm kích hoạt và yk là tín hiệu đầu ra của nơron có nhãn k.

Cách thức kết nối các nơron trong mạng xác định kiến trúc (topology) của mạng. Các nơron trong mạng có thể kết nối đầy đủ (fully connected) tức là mỗi nơron đều được kết nối với tất cả các nơron khác, hoặc kết nối cục bộ (partially connected) chẳng hạn chỉ kết nối giữa các nơron trong các tầng khác nhau.

***2.1.3 Một số hàm kích hoạt phổ biến***

**a) Sigmoid**

Công thức:

Phân tích: Hàm Sigmoid nhận đầu vào là một số thực và chuyển thành một giá trị trong khoảng (0;1). Đầu vào là số thực âm rất nhỏ sẽ cho đầu ra tiệm cận với 0, ngược lại, nếu đầu vào là một số thực dương lớn sẽ cho đầu ra là một số tiệm cận với 1. Trong quá khứ hàm Sigmoid hay được dùng vì có đạo hàm rất đẹp. Tuy nhiên hiện nay hàm Sigmoid rất ít được dùng vì những nhược điểm sau: Hàm Sigmoid bão hào và triệt tiêu gradient, hơn nữa không có trung tâm là 0 gây khó khăn cho việc hội tụ.

Chart, line chart

Description automatically generated

Hình 1.13. Đồ thị hàm Sigmoid

**b) Tanh**

Công thức:

Phân tích:

Chart, line chart

Description automatically generated

Hình 1.14. Đồ thị hàm Tanh

Hàm Tanh nhận đầu vào là một số thực và chuyển thành một giá trị trong khoảng (-1; 1). Cũng như Sigmoid, hàm Tanh bị bão hoà ở 2 đầu (gradient thay đổi rất ít ở 2 đầu). Tuy nhiên hàm Tanh lại đối xứng qua 0 nên khắc phục được một nhược điểm của Sigmoid.

**c) ReLU**

Công thức:

Phân tích: Hàm ReLU đang được sử dụng khá nhiều trong những năm gần đây khi huấn luyện các mạng Neural. ReLU đơn giản lọc các giá trị < 0. Một số ưu điểm khá vượt trội của nó so với Sigmoid và Tanh:

* Tốc độ hội tụ nhanh hơn hẳn.
* Tính toán nhanh hơn.

Tuy nhiên ReLU cũng có một nhược điểm:

* Với các node có giá trị nhỏ hơn 0, qua ReLU activation sẽ thành 0, hiện tượng đấy gọi là “Dying ReLU“. Nếu các node bị chuyển thành 0 thì sẽ không có ý nghĩa với bước linear activation ở lớp tiếp theo và các hệ số tương ứng từ node đấy cũng không được cập nhật với gradient descent. Do đó Leaky ReLU ra đời.
* Khi learning rate lớn, các trọng số (weights) có thể thay đổi theo cách làm tất cả neuron dừng việc cập nhật.

Chart, line chart

Description automatically generated

Hình 1.15. Đồ thị hàm ReLU

**c) Leaky ReLU**

Công thức:

Phân tích:

Leaky ReLU là một cố gắng trong việc loại bỏ "dying ReLU". Thay vì trả về giá trị 0 với các đầu vào < 0 thì Leaky ReLU tạo ra một đường xiên có độ dốc nhỏ (xem đồ thị). Có nhiều báo cáo về việc hiệu Leaky ReLU có hiệu quả tốt hơn ReLU, nhưng hiệu quả này vẫn chưa rõ ràng và nhất quán.

Ngoài Leaky ReLU có một biến thể cũng khá nổi tiếng của ReLU là PReLU. PReLU tương tự Leaky ReLU nhưng cho phép neural tự động chọn hệ số α tốt nhất.

Chart, line chart

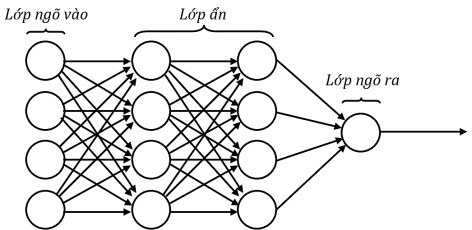
Description automatically generated

Hình 1.16. Đồ thị hàm Leaky ReLU

***2.1.4 Multi-layers Perceptron (MLP)***

Mô hình mạng nơron được sử dụng rộng rãi nhất là mô hình mạng nhiều lớp truyền thẳng (MLP: Multi Layer Perceptron). Mạng Neural truyền thẳng là mạng neural mà mỗi neuron lớp sau đều được nối với tất cả các neuron lớp trước. Một mạng MLP tổng quát là mạng có n (n≥2) lớp (thông thường lớp đầu vào không được tính đến): trong đó gồm một lớp đầu ra (lớp thứ n) và (n-1) lớp ẩn, bao gồm:

* Lớp ngõ vào (input layer), cung cấp dữ liệu từ ngoài vào mạng.
* Lớp ngõ ra (output layer) có nhiệm vụ thực hiện tính toán và truyền thông tin ra bên ngoài.
* Lớp ẩn (hidden layer) thực hiện nhận thông tin từ các lớp phía trước, tính toán và chuyển thông tin đến các lớp phía sau.



Hình 1.17. Cấu trúc mạng đa lớp MLP

1. https://nhandan.vn/baothoinay-dothi/tim-co-che-cho-cac-diem-do-xe-637883/ [↑](#footnote-ref-1)
2. Facebook, Inc. là một công ty truyền thông xã hội và công nghệ Mỹ có trụ sở tại Menlo Park, California. [↑](#footnote-ref-2)
3. TensorFlow là một thư viện phần mềm mã nguồn mở dành cho máy học trong nhiều loại hình tác vụ nhận thức và hiểu ngôn ngữ. [↑](#footnote-ref-3)
4. Keras là một thư viện phần mềm mã nguồn mở cung cấp giao diện Python cho các mạng nơ-ron nhân tạo. [↑](#footnote-ref-4)
5. Alan Mathison Turing OBE FRS là một nhà toán học, logic học và mật mã học người Anh, được xem là cha đẻ của ngành khoa học máy tính. [↑](#footnote-ref-5)
6. Pinterest là website chia sẻ ảnh theo dạng mạng xã hội [↑](#footnote-ref-6)
7. Facebook là một mạng xã hội phổ biến nhất hiện nay [↑](#footnote-ref-7)
8. Amazon.com, Inc. là một công ty công nghệ đa quốc gia của Mỹ có trụ sở tại Seattle, Washington [↑](#footnote-ref-8)
9. AlphaGo là chương trình máy tính cờ vây do Google DeepMind phát triển tại London [↑](#footnote-ref-9)
10. Matplotlib là một thư viện vẽ đồ thị cho ngôn ngữ lập trình Python và phần mở rộng toán học số NumPy của nó [↑](#footnote-ref-10)
11. Pillow là một phần của PIL- Python Image Library, thư viện xử lý ảnh của Python [↑](#footnote-ref-11)
12. OpenCV là một thư viện các chức năng lập trình chủ yếu nhắm vào thị giác máy tính thời gian thực [↑](#footnote-ref-12)
13. Số liệu lấy từ https://insights.stackoverflow.com/survey/2020 [↑](#footnote-ref-13)
14. nVIDIA, một tập đoàn đa quốc gia, chuyên về phát triển bộ xử lý đồ họa và công nghệ chipset cho các máy trạm, máy tính cá nhân, và các thiết bị di động [↑](#footnote-ref-14)
15. Amazon Web Services là một công ty con của Amazon cung cấp các nền tảng điện toán đám mây theo yêu cầu cho các cá nhân, công ty và chính phủ, trên cơ sở trả tiền theo nhu cầu sử dụng. [↑](#footnote-ref-15)
16. Google Cloud Platform, được cung cấp bởi Google, là một bộ dịch vụ điện toán đám mây chạy trên cùng một cơ sở hạ tầng mà Google sử dụng nội bộ cho các sản phẩm của người dùng cuối, như Google Search và YouTube. [↑](#footnote-ref-16)