TRƯỜNG ĐẠI HỌC KINH TẾ QUỐC DÂN

**Năm học: 2025. Kỳ học: I CÔNG NGHỆ THÔNG TIN Nguyễn Huy Hoàng 11218458**

**TRƯỜNG CÔNG NGHỆ**

--------🙞🙞🕮🙜🙜--------



**KHÓA LUẬN TỐT NGHIỆP**

**Ứng dụng di động ICarP và hệ thống bãi gửi xe thông minh ICarP**

**Tên sinh viên : Nguyễn Huy Hoàng**

**Giảng viên hướng dẫn : TS. Phạm Minh Hoàn**

**Hà Nội, 4/2025**

**TRƯỜNG ĐẠI HỌC KINH TẾ QUỐC DÂN   
TRƯỜNG CÔNG NGHỆ**--------🙞🙞🕮🙜🙜--------



**KHÓA LUẬN TỐT NGHIỆP**

**Tên đề tài**

**Tên sinh viên : Nguyễn Huy Hoàng**

**Mã sinh viên : 11218458**

**Lớp : Công nghệ thông tin K63**

**Ngành : Công nghệ thông tin**

**Khoa : Công nghệ thông tin**

**Giảng viên hướng dẫn : TS. Phạm Minh Hoàn**

*(Chữ ký GVHD)*

**Hà Nội, 4/2025**

MỤC LỤC

[LỜI CAM ĐOAN iv](#_Toc191523833)

[LỜI CẢM ƠN v](#_Toc191523834)

[DANH MỤC CÁC TỪ VIẾT TẮT vi](#_Toc191523835)

[DANH MỤC BẢNG, BIỂU ĐỒ vii](#_Toc191523836)

[DANH MỤC HÌNH viii](#_Toc191523837)

[MỞ ĐẦU 1](#_Toc191523838)

[CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI 3](#_Toc191523839)

[1.1. Lý do chọn đề tài. 3](#_Toc191523840)

[1.2. Mục tiêu của đề tài. 4](#_Toc191523841)

[1.3. Đối tượng nghiên cứu. 5](#_Toc191523842)

[1.3.1. Khái niệm đối tượng. 5](#_Toc191523843)

[1.3.2. Đối tượng hệ thống nghiên cứu. 5](#_Toc191523844)

[1.3.2.1. Đối tượng – Phương tiện giao thông ô tô 5](#_Toc191523845)

[1.3.2.2. Đối tượng – Người dùng hệ thống/Chủ sở hữu phương tiện. 6](#_Toc191523846)

[1.4. Phương pháp nghiên cứu. 8](#_Toc191523847)

[1.4.1. Phương pháp nghiên cứu tài liệu - công nghệ 8](#_Toc191523848)

[1.4.2. Phương pháp phân tích và thiết kế hệ thống. 9](#_Toc191523849)

[1.4.3. Phương pháp cài đặt và kiểm thử. 9](#_Toc191523850)

[1.5. Quy trình nghiệp vụ của hệ thống. 10](#_Toc191523851)

[1.5.1. Quy trình sử dụng của ứng dụng di động ICarP. 10](#_Toc191523852)

[1.5.1.1. Các bước sử dụng ứng dụng. 10](#_Toc191523853)

[1.5.1.2. Các chức năng khác. 14](#_Toc191523854)

[1.5.2. Quy trình xử lý của hệ thống bãi gửi xe ICarP. 14](#_Toc191523855)

[1.5.2.1. Check – in. 14](#_Toc191523856)

[1.5.2.2. Check – out. 15](#_Toc191523857)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 16](#_Toc191523858)

[2.1. Phần mềm lập trình Visual Studio Code 16](#_Toc191523859)

[2.1.1. Giới thiệu 16](#_Toc191523860)

[2.1.2. Các tính năng nổi bật của Visual Studio Code 16](#_Toc191523861)

[2.1.3. Các tiện ích mở rộng – Extentions được sử dụng 18](#_Toc191523862)

[2.2. Ngôn ngữ lập trình Python 22](#_Toc191523863)

[2.2.1. Giới thiệu 22](#_Toc191523864)

[2.2.2. Đặc điểm của ngôn ngữ 23](#_Toc191523865)

[2.2.3. Ứng dụng của Python 25](#_Toc191523866)

[2.2.4. Các thư viện Python được sử dụng 27](#_Toc191523867)

[2.2.4.1. Thư viện Tkinter 27](#_Toc191523868)

[2.2.4.2. Thư viện OpenCV 29](#_Toc191523869)

[2.2.4.3. Thư viện EasyOCR 30](#_Toc191523870)

[2.2.4.4. Thư viện Numpy 32](#_Toc191523871)

[2.2.4.5. Thư viện RE 33](#_Toc191523872)

[2.3. You Only Live Once – YOLO 35](#_Toc191523873)

[2.3.1. CNN – Convolutional Neural Network 35](#_Toc191523874)

[2.3.1.1. Giới thiệu chung 35](#_Toc191523875)

[2.3.1.2. Đặc điểm chính của CNNs 36](#_Toc191523876)

[2.3.1.3. Cấu trúc của CNNs 36](#_Toc191523877)

[2.3.1.4. Nguyên lý hoạt động của CNNs 37](#_Toc191523878)

[2.3.2. Cấu trúc của YOLO 40](#_Toc191523879)

[2.3.2. Quy trình nhận diện của YOLO 41](#_Toc191523880)

[2.3.3. Công thức tính toán trong YOLO 44](#_Toc191523881)

[2.3.3.1. Công thức tính Confidence Score 44](#_Toc191523882)

[2.3.3.2. Hàm mất mát (Loss Function) 44](#_Toc191523883)

[2.3.3.3. Hàm MSE (Mean Square Error) 44](#_Toc191523884)

[2.4. React Native 45](#_Toc191523885)

[2.5. Thiết bị phần cứng 45](#_Toc191523886)

[CHƯƠNG 3. PHÂN TÍCH – THIẾT KẾ HỆ THỐNG 46](#_Toc191523887)

[3.1. Phân tích chức năng hệ thống 46](#_Toc191523888)

[3.1.1. Chức năng của ứng dụng di động ICarP 46](#_Toc191523889)

[3.2.1. Ứng dụng di động ICarP 47](#_Toc191523890)

[3.2.2. Hệ thống bãi gửi xe thông minh ICarP 49](#_Toc191523891)

[KẾT LUẬN 51](#_Toc191523892)

[TÀI LIỆU THAM KHẢO 52](#_Toc191523893)

[PHỤ LỤC 53](#_Toc191523894)

[Kết quả turnitin 55](#_Toc191523895)

# LỜI CAM ĐOAN

*Hà nội, ngày tháng năm 20*

**Sinh viên**

**(ký, ghi rõ họ tên)**

# LỜI CẢM ƠN

*(Nếu có)*

# DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **Từ viết tắt** | **Ý nghĩa** |
| API | Giao diện lập trình ứng dụng  (Application Programming Interface) |
| EUD | Phát triển ứng dụng người dùng cuối  (End-User Development) |

# DANH MỤC BẢNG, BIỂU ĐỒ

*(Bảng, biểu đồ được đánh số thứ tự theo Chương: ví dụ bảng 1.1 là bảng đầu tiên nằm trong chương 1, Biểu đồ 2.1 là biểu đồ đầu tiên nằm trong chương 2; Liệt kê hết bảng sau đó đến các biểu đồ)*

Bảng 1.1: Tên nội dung bảng 1

Bảng 2.1: Tên nội dung bảng 2

Bảng 3.1: Tên nội dung bảng 3

……

Biểu đồ 2.1: Tên nội dung biểu đồ 5

# DANH MỤC HÌNH

*(Hình đánh số thứ tự theo Chương: ví dụ hình 1.1 hình đầu tiên trong chương 1, hình 2.1 là hình đầu tiên trong chương 2)*

Hình 1.1: Tên nội dung hình 1

Hình 2.1: Tên nội dung hình 2

Hình 3.1: Tên nội dung hình 3

# MỞ ĐẦU

Trong bối cảnh đô thị hóa ngày càng phát triển, nhu cầu di chuyển và sở hữu ô tô của người dân cũng gia tăng mạnh mẽ. Tuy nhiên, việc tìm kiếm bãi đậu xe ô tô và quản lý bãi đỗ hiệu quả vẫn là một vấn đề nan giải đối với nhiều thành phố, đặc biệt là tại các khu vực trung tâm với mật độ giao thông dày đặc. Em nhận thấy rằng nhiều người lái xe phải tốn nhiều thời gian và công sức để tìm kiếm chỗ đậu xe, gây nên tình trạng ùn tắc giao thông và ảnh hưởng đến hiệu suất công việc. Hơn nữa, nhiều bãi đỗ xe vẫn sử dụng phương pháp quản lý thủ công, gây lãng phí diện tích, nhân công, thời gian và không tận dụng được hết công suất của bãi gửi xe.

Sự phát triển mạnh mẽ của công nghệ, đặc biệt là các công nghệ đa nền tảng như React Native cho Mobile App Developement, Python cùng các thư viện đa dạng cho việc xây dựng Desktop Application và việc xử lý và quản lý dữ liệu được đơn giản quá và tăng hiệu quả nhờ vào SQL Server. Nhờ những công nghệ trên tạo điều kiện thuận lợi để xây dựng một hệ thống gửi xe tự động thông minh, tự động và xây dựng một ứng dụng điện thoại di động trong cùng môi trường. Hệ thống này không chỉ giúp người dùng dễ dàng tìm kiếm và đặt chỗ gửi xe theo thời gian thực mà còn hỗ trợ nhận diện biển số bằng công nghệ AI, tối ưu hóa quy trình vào/ra bãi xe mà không cần đến nhân công thực tế. Hơn thế nữa, hệ thống cung cấp các báo cáo thống kê chi tiết về lưu lượng xe ra vào, hỗ trợ quản lý và vận hành bãi đỗ một cách hiệu quả, tiết kiệm.

Mục tiêu của đề tài là phát triển một hệ thống bãi gửi xe thông minh, giúp người dùng dễ dàng tìm kiếm, đặt chỗ và quản lý việc gửi xe một cách hiệu quả. Hệ thống cần đảm bảo khả năng hoạt động đa nền tảng, hiệu suất cao, bảo mật tốt và tích hợp các công nghệ hiện đại như nhận diện biển số xe bằng AI, thanh toán tự động và theo dõi thời gian gửi xe theo thời gian thực. Người dùng có thể nhanh chóng đăng ký, đăng nhập, quản lý thông tin phương tiện, kiểm tra tình trạng bãi đỗ và tìm kiếm bãi gửi xe phù hợp với nhu cầu, vị trí. Đặc biệt, AI sẽ hỗ trợ phân tích lưu lượng xe, tối ưu hóa quy trình vận hành và đưa ra các gợi ý nhằm nâng cao trải nghiệm người dùng cũng như tối đa hóa hiệu suất sử dụng bãi đỗ.

Bài báo cáo có bố cục gồm 5 chương:

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

Trình bày lý do chọn đề tài, mục đích, đối tượng và phạm vi, ý nghĩa khoa học và thực tiễn của đề tài. Chỉ ra những vấn đề mà đề tài tập trung nghiên cứu giải quyết.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Trình bày về các công nghệ được sử dụng trong hệ thống như Python cùng các thư viện, công nghệ React Native trong việc phát triển ứng dụng di động và model học máy YOLO trong việc xử lý hình ảnh phương tiện. Giới thiệu về các thiết bị phần cứng được sử dụng trong hệ thống.

CHƯƠNG 3: PHÂN TÍCH – THIẾT KẾ HỆ THỐNG

Trình bày về các chức năng của hệ thống như đăng ký, đăng nhập, tìm kiếm bãi gửi xe, nhận diện xe/biển số, quét mã QR tự động cùng các biểu đồ về hệ thống như biểu đồ hoạt động, biểu đồ đặc tả use case, .v.v.

Trình bày về phần thiết kế giao diện hệ thống, giao diện người dùng trên ứng dụng di động. Thiết kế cơ sở dữ liệu phù hợp, đáp ứng yêu cầu về lưu trữ thông tin và thuận tiện cho việc phân tích và đưa ra dự đoán từ bộ dữ liệu.

CHƯƠNG 4: CÀI ĐẶT – KIỂM THỬ PHẦN MỀM

Trình bày quá trình xây dựng hệ thống theo yêu cầu. Tiến hành cài đặt hệ thống trên thiết bị thực tế và kiểm thử các chức năng của hệ thống.

CHƯƠNG 5: KẾT LUẬN – ĐÁNH GIÁ PHẦN MỀM

Trình bày về kết quả đạt được của hệ thống, các tối ưu và các thiếu sót trong quá trình sử dụng. Đánh giá tổng quan về hệ thống và đưa ra các phương hướng xây dựng và phát triển hệ thống sau này.

Thông qua đề tài này, sinh viên sẽ có cơ hội nghiên cứu và áp dụng các công nghệ hiện đại vào việc xây dựng một hệ thống bãi gửi xe thông minh hoàn chỉnh. Báo cáo sẽ cung cấp cái nhìn tổng quan về toàn bộ quá trình phát triển hệ thống, từ việc phân tích yêu cầu, thiết kế kiến trúc, xây dựng cơ sở dữ liệu, lập trình frontend và backend, đến triển khai và đánh giá hiệu suất. Đồng thời, đề tài cũng giúp sinh viên học cách tích hợp các công nghệ như nhận diện biển số bằng AI, thanh toán tự động và tối ưu hóa quy trình vận hành. Qua đó, sinh viên không chỉ nâng cao khả năng giải quyết vấn đề thực tế mà còn có thể ứng dụng những kiến thức này vào việc phát triển các hệ thống thông minh khác trong lĩnh vực công nghệ phần mềm trong tương lai.

# CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI

## 1.1. Lý do chọn đề tài.

Trong bối cảnh đô thị hóa ngày càng phát triển, nhu cầu di chuyển và sở hữu ô tô của người dân cũng gia tăng mạnh mẽ. Tuy nhiên, việc tìm kiếm bãi đậu xe ô tô và quản lý bãi đỗ hiệu quả vẫn là một vấn đề nan giải đối với nhiều thành phố, đặc biệt là tại các khu vực trung tâm với mật độ giao thông dày đặc. Em nhận thấy rằng nhiều người lái xe phải tốn nhiều thời gian và công sức để tìm kiếm chỗ đậu xe, gây nên tình trạng ùn tắc giao thông và ảnh hưởng đến hiệu suất công việc.

Hiện nay, trên thị trường có nhiều bãi gửi xe phục vụ nhu cầu của người dân, nhưng phần lớn vẫn hoạt động theo phương thức thủ công. Các bãi đỗ xe truyền thống chủ yếu dựa vào nhân viên trông giữ để ghi vé, kiểm soát lượt xe ra vào và thu phí trực tiếp. Điều này không chỉ tốn nhiều thời gian mà còn dễ dẫn đến sai sót, thất thoát doanh thu và ùn tắc vào các khung giờ cao điểm. Bên cạnh đó, việc thiếu hệ thống quản lý thông minh khiến người dùng gặp khó khăn trong việc tìm kiếm chỗ đỗ, theo dõi thời gian gửi xe và thanh toán linh hoạt.

Chính vì vậy, nhu cầu về một hệ thống gửi xe tự động, hiện đại, cho phép người dùng tìm kiếm bãi đỗ xe phù hợp giựa trên vị trí hiện tại kết hợp với hệ thống bãi gửi xe thông minh, hoàn toàn tự động là vô cùng hợp lý và cần thiết. Với một hệ thống như thế, chúng ta không chỉ giải quyết vấn đề tìm chỗ đậu nhanh chóng và dễ dàng cho người lái xe mà còn góp phần cải thiện tình trạng giao thông đô thị. Hệ thống thông minh này sẽ giúp tối ưu hóa việc sử dụng không gian, giảm thiểu ùn tắc, tiết kiệm thời gian cho người sử dụng và giảm thiểu lượng khí thải phát sinh từ việc di chuyển tìm chỗ đậu xe. Đồng thời, việc ứng dụng công nghệ vào việc quản lý bãi đỗ xe cũng nâng cao hiệu quả hoạt động và mang lại tiện ích lớn cho cả người sử dụng lẫn chủ sở hữu bãi đỗ. Hơn thế nữa, việc phát triển các hệ thống tự động, thông minh và chính xác còn là một bước tiến rất quan trọng trong quá trình xây dựng đất nước giàu mạnh và hiện đại.

Chính vì những lý do trên, em chọn đề tài này để phát triển ‘ICarP’ – Intelligent Car Parking một giải pháp công nghệ hiện đại đạt được tất cả các yêu cầu trên. Nhanh, mạnh, chính xác và thông minh chính là những gì ‘ICarP’ có để phục vụ người dùng, chung tay góp phần cải thiện chất lượng sống và hướng đến một xã hội văn minh.

## 1.2. Mục tiêu của đề tài.

Ứng dụng và hệ thống ‘ICarP’ được phát triển nhằm giải quyết vấn đề đỗ xe tại các khu đô thị đông đúc. Ứng dụng giúp người dùng nhanh chóng tìm kiếm bãi gửi xe gần nhất, kiểm tra tình trạng chỗ trống theo thời gian thực, di chuyển đến bãi đỗ phù hợp theo lộ trình đường đi có sẵn, tiết kiệm thời gian và giảm căng thẳng khi tìm nơi đỗ xe.

Hệ thống bãi gửi xe thông minh sử dụng công nghệ cảm biến, camera và trí tuệ nhân tạo để tối ưu hóa việc quản lý, giúp tăng hiệu suất sử dụng bãi xe, giảm ùn tắc và nâng cao trải nghiệm người dùng. Hệ thống tự động còn giúp giảm thiểu chi phí thuê nhân công thực tế, giảm thiếu sót và sai lệch từ con người. Nhờ đó, quá trình gửi và lấy xe trở nên nhanh chóng, thuận tiện hơn, hạn chế tối đa tình trạng chờ đợi và ùn tắc, đặc biệt vào các khung giờ cao điểm. Hệ thống còn có khả năng nhận diện biển số xe tự động, hướng dẫn người dùng đến vị trí đỗ phù hợp và hỗ trợ thanh toán không tiền mặt, giúp tiết kiệm thời gian và tăng cường tính minh bạch trong quản lý.

Ngoài ra, dữ liệu thu thập từ hệ thống sẽ được phân tích để dự đoán xu hướng sử dụng bãi xe, tối ưu hóa phân bổ không gian đỗ và đề xuất các giải pháp cải thiện hiệu suất vận hành. Nhờ ứng dụng công nghệ hiện đại, hệ thống bãi gửi xe thông minh không chỉ mang lại lợi ích cho người dùng mà còn hỗ trợ các đơn vị quản lý nâng cao hiệu quả kinh doanh và vận hành bãi đỗ một cách tối ưu.

## 1.3. Đối tượng nghiên cứu.

### 1.3.1. Khái niệm đối tượng.

Trong lập trình, đối tượng (object) là một thực thể có trạng thái (dữ liệu) và hành vi (phương thức). Đối tượng trong lập trình có các đặc điểm sau:

- Thuộc tính (Attributes/Properties): Là thông tin hoặc dữ liệu của đối tượng được lưu thông qua các biến – còn gọi là trường hoặc field.

- Phương thức (Methods/Behaviors): Là các hành động mà đối tượng có thể thực hiện, được định nghĩa dưới dạng các hàm (Functions) trong lớp (Class).

### 1.3.2. Đối tượng hệ thống nghiên cứu.

Đối tượng mà hệ thống bãi gửi xe thông minh ICarP nghiên cứu bao gồm 2 nhóm đối tượng chính:

Đối tượng – Phương tiện giao thông ô tô

Đối tượng – Người dùng hệ thống/Chủ sở hữu phương tiện

#### 1.3.2.1. Đối tượng – Phương tiện giao thông ô tô

*Phương diện:* Tính chất hình học.

Ô tô thường có dạng hộp chữ nhật hoặc hình lăng trụ khi xét về kích thước tổng thể. Dưới dạng mô hình 3D, có thể biểu diễn ô tô bằng một khối hộp chữ nhật (cuboid) với các thông số chiều dài, chiều rộng và chiều cao. Các đối tượng ô tô thường có tính chất hình học khá tương đồng giữa các loại xe cụ thể.

Kích thước cơ bản của một số loại xe như sau:

* Xe con (hatchback, sedan): (~4.0 - 4.5m dài, ~1.7 - 1.8m rộng, ~1.4 - 1.5m cao).
* Xe SUV, crossover: (~4.5 - 5.0m dài, ~1.8 - 2.0m rộng, ~1.6 - 1.8m cao).
* Xe bán tải, xe tải nhỏ: (~5.0 - 6.0m dài, ~2.0 - 2.5m rộng, ~1.7 - 2.5m cao).
* Xe buýt, xe tải lớn: (>6m dài, >2.5m rộng, >2.5m cao).

*Phương diện:* Đặc điểm kết cấu và phân loại.

Cũng như tính chất về hình học, các đôi tượng ô tô cũng chia sẻ chung một đặc điểm về kết cấu giữa các loại xe khác nhau.

*Phương diện:* Biển số xe

Biển số xe của mỗi xe là độc nhất. Biển số chứa thông tin về xuất xứ, khu vực, loại xe, .v.v.

#### 1.3.2.2. Đối tượng – Người dùng hệ thống/Chủ sở hữu phương tiện.

*Phương diện:* Thông tin cá nhân.

Trong hệ thống bãi gửi xe thông minh, đối tượng người dùng chủ yếu là chủ xe hoặc tài xế, những người có nhu cầu gửi xe tại bãi đỗ. Để phục vụ tốt hơn, hệ thống cần thu thập và quản lý thông tin cá nhân của họ một cách hiệu quả và bảo mật.

*Họ và tên:* Dùng để xác định danh tính người sử dụng hệ thống.

*Số điện thoại:* Dùng để xác nhận tài khoản, nhận thông báo về thời gian gửi xe, thanh toán và nhắc nhở khi sắp hết thời gian đỗ xe.

*Email:* Dùng để đăng ký tài khoản, khôi phục mật khẩu và nhận hóa đơn điện tử.

*Biển số xe:* Hệ thống sử dụng camera AI để nhận diện xe ra vào.

*Loại xe:* Sedan, SUV, xe máy, xe tải,... giúp phân loại và sắp xếp chỗ đỗ phù hợp.

*Màu sắc, hãng xe:* Có thể giúp nhận diện xe nhanh chóng khi có sự cố hoặc tìm xe trong bãi.

*Lịch sử thanh toán:* Ghi nhận các giao dịch gửi xe, giúp người dùng kiểm soát chi tiêu.

*Phương thức thanh toán:* Ví điện tử, thẻ ngân hàng, tiền mặt,... để hỗ trợ thanh toán nhanh chóng.

*Chương trình ưu đãi:* Thẻ thành viên, điểm thưởng hoặc gói đăng ký gửi xe theo tháng.

*Tần suất gửi xe:* Xác định mức độ sử dụng bãi đỗ để cá nhân hóa dịch vụ.

*Khung giờ thường xuyên gửi xe:* Dùng để tối ưu hóa bố trí chỗ đỗ và dự báo tình trạng quá tải.

*Bãi đỗ xe ưa thích:* Hệ thống có thể gợi ý chỗ đỗ dựa trên lịch sử đỗ xe của người dùng.

Việc nghiên cứu người dùng trên phương diện thông tin cá nhân giúp hệ thống gửi xe thông minh hoạt động hiệu quả hơn, cá nhân hóa trải nghiệm cho khách hàng và tăng cường bảo mật. Hệ thống có thể tối ưu hóa quy trình đỗ xe, hỗ trợ thanh toán linh hoạt và giúp bãi xe vận hành thông minh hơn, đáp ứng tốt nhu cầu của người dùng.

*Phương diện:* Phân loại người lái xe theo mục đích.

Trong hệ thống bãi gửi xe thông minh, việc phân loại người lái xe theo mục đích sử dụng phương tiện giúp tối ưu hóa trải nghiệm người dùng, quản lý chỗ đỗ hiệu quả và điều chỉnh chính sách giá phù hợp.

*Người lái xe cá nhân:* Sử dụng xe cho mục đích cá nhân hoặc gia đình. Thường gửi xe tại nhà, nơi làm việc hoặc trung tâm mua sắm. Ưu tiên bãi đỗ an toàn, tiện lợi, có dịch vụ trông giữ lâu dài. Có thể sử dụng bãi gửi xe theo ngày, theo giờ hoặc vé tháng.

*Người lái xe dịch vụ:* Xe thường xuyên di chuyển, dừng đỗ trong thời gian ngắn. Ưu tiên bãi đỗ có giá rẻ, thanh toán nhanh và không mất nhiều thời gian đỗ xe. Có thể cần khu vực đón/trả khách thuận tiện. Hệ thống có thể áp dụng chính sách ưu tiên, giảm giá cho tài xế công nghệ.

*Người lái xe thương mại:* Thường di chuyển bằng xe sang, xe công vụ hoặc xe cho thuê dài hạn. Cần bãi đỗ xe an toàn, có hệ thống bảo vệ tốt. Ưu tiên các bãi xe có dịch vụ VIP, đỗ xe có mái che, có bảo hiểm trông giữ. Có thể sử dụng dịch vụ đặt chỗ trước hoặc hợp đồng đỗ xe dài hạn.

*Người lái xe giao hàng, vận tải nhỏ:* Sử dụng xe tải nhỏ, xe van hoặc xe bán tải. Cần bãi đỗ gần kho hàng, trung tâm thương mại, siêu thị. Đỗ xe trong thời gian ngắn để giao nhận hàng, có thể cần chỗ đỗ ưu tiên. Cần quy trình ra vào nhanh, có hỗ trợ đăng ký trước.

*Người lái xe du lịch, xe đường dài:* Sử dụng xe cá nhân hoặc xe thuê để di chuyển đường dài. Cần bãi đỗ xe có thời gian gửi dài (qua đêm, nhiều ngày). Ưu tiên bãi đỗ gần khu du lịch, khách sạn, sân bay. Cần các dịch vụ bổ trợ như rửa xe, sạc điện (đối với xe điện), bảo dưỡng nhanh.

*Người lái xe điện:* Sử dụng xe chạy bằng pin, cần có trạm sạc điện tại bãi đỗ. Ưu tiên bãi đỗ có khu vực dành riêng cho xe điện. Cần thời gian đỗ lâu hơn để sạc đầy pin.

Việc phân loại người lái xe theo mục đích sử dụng phương tiện giúp hệ thống bãi gửi xe thông minh hoạt động hiệu quả hơn, tăng sự tiện lợi cho người dùng và tối ưu hóa việc quản lý bãi xe.

## 1.4. Phương pháp nghiên cứu.

Đề tài áp dụng nhiều phương pháp nghiên cứu, phân tích, thiết kế, xây dựng phù hợp nhằm phát triển hệ thống đúng với yêu cầu đặt ra.

### 1.4.1. Phương pháp nghiên cứu tài liệu - công nghệ

Bước đầu tiên của quá trình xây dựng hệ thống là nghiên cứu đề tài, bao gồm tìm hiểu về các hệ thống bãi gửi xe truyền thống, đánh giá ưu nhược điểm của chúng và phân tích các công nghệ hiện đại có thể áp dụng để tối ưu hóa quy trình quản lý bãi đỗ xe.

Hệ thống bãi gửi xe truyền thống chủ yếu hoạt động dựa trên sự quản lý thủ công của nhân viên bãi xe, sử dụng vé giấy, thẻ từ hoặc ghi chép sổ sách để kiểm soát phương tiện ra vào. Bãi gửi xe truyền thống có một số ưu điểm như: Dễ triển khai, không yêu cầu công nghệ hiện đại, phức tạp, phù hợp với các bãi gửi xe nhỏ lẻ. Cùng với đó là các nhược điểm như: Thời gian xử lý chậm, dễ xảy ra sai sót, khó kiểm soát và tối ưu.

Nếu áp dụng các công nghệ hiện đại thì hệ thống bãi gửi xe thông minh có thể giải quyết được các nhược điểm trên thông qua việc ứng dụng các model trí tuệ nhân tạo trong việc xử lý xe, ứng dụng di động giúp người lái xe dễ dàng thao tác và tìm kiếm bãi gửi xe, .v.v.

Bên cạnh đó, để có thể áp dụng các công nghệ hiện đại thì chúng ta cũng phải tìm hiểu sâu về các công nghệ hiện có và tìm ra lối đi phù hợp giúp việc xây dựng hệ thống nhanh gọn và hiệu quả.

*React Native*: Phát triển ứng dụng di động đa nền tảng, giúp việc xây dựng ứng dụng di động nhanh, ít lỗi, hiệu quả và bảo trì ứng dụng dễ dàng.

*Python*: Python là một ngôn ngữ lập trình thế hệ mới với bộ công cụ dễ dàng tiếp cận, nhanh và mạnh. Cùng với đó là cộng đồng các lập trình viên sử dụng ngôn ngữ này rất lớn, giúp cho việc tìm hiểu khi xảy ra vấn đề vô cùng thuận tiện. Với hệ thống các thư viện mở đồ sộ, Python là ứng cử viên sáng giá trong việc xây dựng một hệ thống thông minh, tự động hoàn toàn.

*YOLO*: YOLO (You Only Look Once) là một thuật toán nhận diện đối tượng theo thời gian thực dựa trên mạng nơ-ron tích chập (CNN - Convolutional Neural Network). YOLO có khả năng phát hiện và phân loại nhiều đối tượng cùng lúc trong một khung hình duy nhất, giúp tối ưu hóa tốc độ xử lý mà vẫn đảm bảo độ chính xác cao.

### 1.4.2. Phương pháp phân tích và thiết kế hệ thống.

Quá trình phân tích và thiết kế hệ thống diễn ra nhằm đảm cảo cấu trúc phần mềm của hệ thống đạt đúng yêu cầu, thuận tiện cho việc mở rộng chức năng hoặc bảo trì trong tương lai.

Hệ thống bao gồm hai ứng dụng chính:

Ứng dụng di động ICarP được thiết kế và xây dựng thông qua công nghệ React Native. Thiết kế UI/UX cũng như Backend đều có thể được đảm nhận bởi React Native.

Hệ thống bãi gửi xe thông minh ICarP được xây dựng chủ yếu bằng ngôn ngữ Python và các thư viện như Tkinter để xử lý Frontend và model học máy YOLO để xử lý hình ảnh.

Các sơ đồ UML được xây dựng để thiết kế hệ thống chặt chẽ, bao gồm:

*Sơ đồ luồng dữ liệu (DFD)* giúp mô tả cách dữ liệu di chuyển giữa các thành phần của hệ thống.

*Sơ đồ quan hệ thực thể (ERD)* giúp thiết kế cơ sở dữ liệu tối ưu và dễ dàng mở rộng.

*Sơ đồ hoạt động (Activity Diagram)* giúp mô tả quy trình nghiệp vụ như tìm kiếm bãi gửi xe, thanh toán, nạp tiền, ra/vào bãi gửi xe.

*Biểu đồ ca sử dụng (Use Case)* …

### 1.4.3. Phương pháp cài đặt và kiểm thử.

*Kiểm thử đơn vị (Unit Testing)*: Kiểm thử các thành phần nhỏ nhất của ứng dụng, như một hàm hoặc một module. Nhằm đảm bảo từng phần hoạt động chính xác trước khi tích hợp vào hệ thống lớn. Công cụ kiểm thử: PyTest.

*Kiểm thử tích hợp (Integration Testing)*: Kiểm thử cách các module hoặc thành phần hoạt động cùng nhau. Đảm bảo dữ liệu được truyền đúng giữa các phần của hệ thống.

*Kiểm thử hệ thống (System Testing*): Kiểm tra toàn bộ ứng dụng như một sản phẩm hoàn chỉnh. Đánh giá tính năng, hiệu suất, bảo mật, khả năng tương thích.

Việc lựa chọn phương pháp kiểm thử phù hợp giúp đảm bảo chất lượng phần mềm, tăng tính ổn định và giảm lỗi trong quá trình sử dụng.

## 1.5. Quy trình nghiệp vụ của hệ thống.

### 1.5.1. Quy trình sử dụng của ứng dụng di động ICarP.

Ứng dụng di động ICarP là một sản phẩm con trong toàn bộ hệ thống bãi gửi xe thông minh ICarP. Ứng dụng cho phép người dùng thực hiện nhiều thao tác khác nhau liên quan đến việc gửi xe như:

* Tìm kiếm bãi gửi xe phù hợp theo vị trí thực và thời gian thực.
* Nạp tiền và thanh toán hóa đơn gửi xe tại bãi nhanh chóng, tiện lợi.
* Dự báo và gợi ý các bãi gửi xe theo dữ liệu người dùng được hệ thống thu thập.
* Báo cáo và góp ý về hệ thống và các vấn đề người dùng gặp phải khi gửi xe.

#### 1.5.1.1. Các bước sử dụng ứng dụng.

* Bước 1: Thiết lập tài khoản người dùng

*Tải xuống ứng dụng từ Google Play/App Store.*

Ứng dụng sẽ được kiểm duyệt và công khai trên các nền tảng download trực tuyến để dễ tiếp cận đối với người sử dụng và tăng thêm sự uy tín cho ứng dụng.

*Đăng ký tài khoản thành viên*.

Đăng ký tài khoản thành viên để hệ thống có thể xác minh thông tin người dùng và đem đến trải nghiệm người dùng tốt nhất.

*Đăng nhập tài khoản thành viên.*

Người dùng tiến hành đăng nhập vào hệ thống để có thể trải nghiệm các dịch vụ hấp dẫn từ hệ thống ICarP

*Điền form xác minh thông tin cá nhân.*

Người dùng hệ thống sẽ được yêu cầu điền form thông tin bao gồm thông tin cá nhân và thông tin phương tiện đăng ký. Ứng dụng sẽ tiến hành sàng lọc để tránh trùng lặp thông tin. Một đối tượng người dùng có thể sở hữu nhiều phương tiện ô tô nhưng một đối tượng phương tiện ô tô chỉ thuộc quyền sở hữu của một đối tượng người dùng. Việc sàng lọc thông tin nhằm tránh xảy ra lỗi trong quá trình xử lý của ứng dụng và cũng giúp việc quản lý thuận tiện hơn, mang đến trải nghiệm người dùng tốt hơn.

* Bước 2: Nạp tiền vào ứng dụng.

*Liên kết tài khoản ngân hàng.*

Người dùng ứng dụng tiến hành liên kết tài khoản ngân hàng nhằm giúp việc thanh toán và nạp tiền diễn ra nhanh chóng và tiện lợi hơn.

*Liên kết ví điện tử..*

Người dùng ứng dụng tiến hành liên kết với ví điện tử cũng nhằm giúp việc thanh toán và nạp tiền diễn ra nhanh chóng và tiện lợi hơn.

Việc liên kết với tài khoản ngân hàng hay ví điện tử còn giúp khách hàng bỏ qua bước nạp tiền vào ứng dụng. Người dùng có thể thanh toán trực tiếp bằng ngân hàng và ví điện tử khi check out, giúp việc thanh toán đảm bảo an toàn và nhanh chóng.

*Nạp tiền thông qua mã QR.*

Người dùng ứng dụng có thể nạp tiền thông qua quét mã QR chuyển tiền trên các ứng dụng ngân hàng hoặc ví, mặc dù đây là phương pháp thanh toán đơn giản nhất nhưng phương pháp này cũng có nhiều bước xử lý thông qua cả ứng dụng ICarP và ứng dụng ngân hàng khác. Điều này có thể dẫn đến sự kéo dài thời gian thanh toán và xảy ra sai sót từ phía người dùng.

* Bước 3: Tìm kiếm bãi gửi xe phù hợp

*Hệ thống hiển thị các bãi gửi xe phù hợp (Số lượng trống, vị trí gần nhất).*

Ứng dụng ICarP sẽ hiện thị các bãi gửi xe phù hợp theo yêu cầu của người dùng. Người dùng có thể lựa chọn tìm kiếm nhanh, hệ thống sẽ lấy dữ liệu vị trí hiện tại để tìm kiếm các bãi gửi xe gần nhất và hiện thị tuyến đường tối ưu nhất.

Người dùng còn có thể lựa chọn tìm kiếm tại các vị trí khác trên bản đồ tùy vào mục đích sử dụng. Ví dụ như người dùng có thể tìm kiếm trước các bãi gửi xe ở gần nhà, gần công ty, gần các quán café, quán ăn trong khi đang ở một vị trí khác.

Ngoài ra, ứng dụng sẽ hiện thị các thông tin như đơn giá, số lượng chỗ trống, độ dài quãng đường, .v.v. khi người dùng tiến hành lựa chọn bãi để gửi xe.

*Hệ thống hiển thị các bãi đỗ xe người dùng đã sử dụng.*

Dựa vào dự liệu người dùng mà hệ thống thu thập được, hệ thống sẽ lưu trữ lại các vị trí bãi gửi xe mà người dùng đã sử dụng nhằm tăng tốc độ chọn lựa của người dùng và tăng trải nghiệm của người dùng.

Như vậy, thời gian đầu hệ thống sẽ tập trung thu thập dữ liệu từ người dùng, tiếp đó tiến hành phân tích và đưa ra gợi ý. Cuối cùng, với bộ dữ liệu người dùng ngày càng lớn thì hệ thống sẽ phân tích và đưa ra các gợi ý ngày càng phù hợp hơn nữa.

*Hệ thống hiển thị đoạn đường di chuyển.*

Với công nghệ React Native, ứng dụng có tích hợp Google Map API vào trong hệ thống, giúp hiển thị đoạn đường di chuyển theo hình thức quen thuộc của Google Map. Người dùng sẽ dễ tiếp cận và dễ làm quen với các tuyến đường. Hơn thế nữa, việc sử dụng Google Map API còn giúp tận dụng được các thuật toán tìm đường tối ưu nhất của Google, từ đây nâng cao hiệu suất và nâng cao trải nghiệm người dùng.

*Hệ thống dự báo tình trạng bãi gửi xe trong thời gian sắp tới.*

Hệ thống sẽ thu thập dữ liệu gửi xe như ngày, giờ, thời gian gửi, số lượng xe gửi để phân tích và đưa ra các báo cáo về tình trạng bãi gửi xe trong khoảng thời gian sắp tới. Các dự báo này có thể là số lượng xe gửi sẽ nhiều/ít vào khoảng thời gian nào đó, .v.v.

* Bước 4: Quét mã check in vào bãi gửi xe.

*Người dùng lựa chọn bãi gửi xe và lựa chọn xe gửi.*

Sau khi tiến hành lựa chọn bãi gửi xe và di chuyển đễ bãi, người dùng tiến hành chọn xe gửi vào bãi để ứng dụng có thể trích xuất thông tin và chuyển dữ liệu thành mã QR cho khách hàng sử dụng để check in.

*Hệ thống cấp mã QR cho người dùng.*

ứng dụng sẽ cấp một mã QR cho người dùng để tiến hàng check in. Mã QR này sẽ chứa thông tin tài khoản và thông tin phương tiện mà khách hàng đã chọn để gửi. Người dùng ứng dụng chỉ cần quét mã QR này vào hệ thống máy tại bãi gửi xe là có thể tiến hành gửi xe.

*Người dùng quét mã vào máy để check in.*

Sau khi người dùng tiến hành quét mã, bãi gửi xe có hệ thống camera có tích hợp trí tuệ nhân tạo để phát hiện xe cũng như trích xuất thông tin biển số ngay tại thời điểm gửi. Hệ thống sẽ đối chiếu thông tin với thông tin từ mã QR của khách hàng để tính toán. Nếu thông tin trùng khớp thì barrel chính sẽ mở. Nếu thông tin sai, hệ thống sẽ yêu cầu khách quét lại mã QR, sau một số lần sai nhất định thì hệ thống sẽ tiến hành mở barrel phụ để khách có thể quay xe ra khỏi bãi.

*Hệ thống cấp ô gửi xe còn trống.*

Sau khi check in thành công, hệ thống sẽ kiểm tra số ô gửi còn trống và gợi ý cho người dùng ô gửi gần nhất.

*Hệ thống hướng dẫn khách đến vị trí đỗ xe.*

Hệ thống lưu trữ dữ liệu các ô gửi xe để có thể đề ra hướng đi gần nhất đến ô gửi xe đó. Hướng dẫn của hệ thống sẽ không phải là tuyến đường như Google Map API trước đó mà là các chỉ dẫn đơn giản hơn như: Rẽ phải/Đi thẳng/Rẽ trái/Cột F-12.

*Hệ thống xác nhận gửi xe thành công.*

Hệ thống sẽ thông báo gửi xe thành công qua ứng dụng di động sau khi người dùng hoàn thành đủ và đúng các thao tác.

* Bước 5: Quét mã QR check out.

*Hệ thống cấp mã QR cho người dùng.*

Ứng dụng thao tác cấp mã tương tự như khi người dùng check in.

*Người dùng tiến hành quét mã check out tại máy.*

Người dùng tiến hành quét mã tương tự như khi check in.

*Hệ thống thực hiện lên hóa đơn và thanh toán tự động.*

*Hệ thống hiển thị thời gian, ngày, giờ đỗ và tổng thanh toán.*

Hệ thống sẽ tính toán hóa đơn dựa vào đơn giá, thời gian gửi xe và hiển thị thông tin lên trên màn hình. Lưu ý rằng đơn giá có thể thay đổi tùy thuộc vào thời gian gửi xe.

Hệ thống sẽ tiến hành thanh toán tự động nếu người dùng đã liên kết tài khoản ngân hàng, ví điện tử. Hoặc hệ thống cũng sẽ thanh toán tự động nếu trong tài khoản trên hệ thống của khách còn đủ tiền. Cuối cùng hệ thống sẽ hiển thị mã QR để khách quét chuyển tiền nếu như hai phương án trên không thể thực hiện được.

*Hệ thống xác nhận check out thành công.*

#### 1.5.1.2. Các chức năng khác.

*Phản hồi/Đóng góp ý kiến.*

Người dùng ứng dụng có thể đóng góp ý kiến về hệ thống hoặc các quy trình xử lý nghiệp vụ của hệ thống thông qua ứng dụng di động ICarP.

*Báo cáo sự cố.*

*Đường dây nóng, tổng đài hỗ trợ trực tiếp.*

Người dùng có thể báo cáo sự cố gặng phải qua ứng dụng hoặc liên hệ tổng đài.

*Tích điểm thành viên.*

Người dùng sử dụng dịch vụ của hệ thống sẽ được tự động tích điểm thành viên dựa trên số tiền đã tiêu cho hệ thống gửi xe.

*Hướng dẫn sử dụng ứng dụng.*

Trong ứng dụng sẽ có mục hướng dẫn sử dụng, chức năng này sẽ hiển thị hướng dẫn cho các chức năng trong ứng dụng ICarP.

### 1.5.2. Quy trình xử lý của hệ thống bãi gửi xe ICarP.

Hệ thống bãi gửi xe ICarP là ứng dụng chính trong toàn bộ hệ thống ICarP. Hệ thống sẽ thực hiện các thao tác cơ bản của một bãi gửi xe thông thường nhưng dưới sự thông minh và tự động hóa hoàn toàn. Hệ thống không cần nhân công tại các điểm gửi xe mà thuần chỉ có tương tác giữa khách hàng và hệ thống.

#### 1.5.2.1. Check – in.

*Tiếp nhận mã QR của khách.*

*Trích xuất và đối chiếu thông tin với hệ thống.*

Sau khi khách hàng quét mã QR vào máy quét, hệ thống sẽ nhận diện và trích xuất thông tin từ mã QR như thông tin cá nhân, thông tin phương tiện gửi …

Đồng thời, hệ thống cũng tiến hành quét và trích xuất thông tin biển số của phương tiện trong thời gian thực nhờ vào các model trí tuệ nhân tạo, model xử lý hình ảnh được tích hợp trong hệ thống. Thông tin biển số sau đó được đối chiếu với thông tin từ mã QR của khách hàng và hệ thống sẽ đưa ra quyết định dựa vào kết quả đối chiếu.

*Xác nhận gửi xe thành công/Thông báo lỗi qua màn hình hiển thị.*

Nếu kết quả đối chiếu thông tin giữa mã QR và dữ liệu biển số hệ thống quét được là trùng khớp, hệ thống tiến hành hiển thị thông báo trên màn hình thiết bị và màn hình ứng dụng ICarP.

Nếu kết quả đối chiếu là không trùng khớp thì hệ thống cũng sẽ hiển thị thông báo và có các yêu cầu và hướng dẫn cụ thể đối với khách hàng.

*Nâng barrel để khách đưa xe vào bãi.*

Hệ thống sẽ nâng barrel chính để phương tiện di chuyển vào bãi nếu khách hàng check in thành công.

Hệ thống sẽ nâng barrel phụ để phương tiện quay đầu rời khỏi bãi gửi xe nếu quá trình check in không thành công. Sau đó khách hàng có thể quay lại tiếp tục quá trình check in trong trường hợp trước đó xảy ra lỗi.

#### 1.5.2.2. Check – out.

*Tiếp nhận mã QR của khách.*

*Trích xuất và đối chiếu thông tin với hệ thống.*

Quá trình diễn ra như khi khách hàng tiến hành check in.

*Thanh toán hóa đơn gửi xe.*

Hệ thống sẽ tính toán hóa đơn dựa vào đơn giá, thời gian gửi xe và hiển thị thông tin lên trên màn hình. Lưu ý rằng đơn giá có thể thay đổi tùy thuộc vào thời gian gửi xe.

Hệ thống sẽ tiến hành thanh toán tự động nếu người dùng đã liên kết tài khoản ngân hàng, ví điện tử. Hoặc hệ thống cũng sẽ thanh toán tự động nếu trong tài khoản trên hệ thống của khách còn đủ tiền. Cuối cùng hệ thống sẽ hiển thị mã QR để khách quét chuyển tiền nếu như hai phương án trên không thể thực hiện được.

*Xuất hóa đơn nếu khách có nhu cầu.*

Nếu khách có nhu cầu xuất hóa đơn để đối chiếu thì chỉ cần thao tác là hệ thống sẽ xuất hóa đơn cho khách.

*Nâng barrel để xe khách di chuyển ra.*

Sau khi check out thành công, hệ thống sẽ tự động nâng barrel chính để khách hàng di chuyển ra khỏi bãi đỗ xe.

# 

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

## 2.1. Phần mềm lập trình Visual Studio Code

### 2.1.1. Giới thiệu

Visual Studio Code (thường gọi tắt là VS Code) là một trình soạn thảo mã nguồn miễn phí, mạnh mẽ và đa nền tảng do Microsoft phát triển. Được ra mắt vào năm 2015, VS Code nhanh chóng trở thành một trong những công cụ phổ biến nhất trong cộng đồng lập trình nhờ vào sự nhẹ nhàng, tốc độ nhanh, cùng với khả năng hỗ trợ nhiều ngôn ngữ lập trình và tiện ích mở rộng phong phú.

VS Code không chỉ đơn thuần là một trình soạn thảo văn bản mà còn hỗ trợ nhiều tính năng nâng cao như debugging, quản lý Git, tích hợp terminal, IntelliSense (gợi ý mã thông minh) và nhiều tiện ích mở rộng giúp tối ưu hóa quy trình lập trình.

### 2.1.2. Các tính năng nổi bật của Visual Studio Code

*Hỗ trợ đa ngôn ngữ lập trình.*

VS Code hỗ trợ rất nhiều ngôn ngữ lập trình khác nhau, bao gồm JavaScript, Python, Java, C/C++, PHP, TypeScript, Go, Ruby, Rust và nhiều ngôn ngữ khác. Điều này giúp lập trình viên có thể làm việc trên nhiều dự án với các công nghệ khác nhau mà không cần chuyển đổi giữa các trình soạn thảo khác nhau. Ngoài ra, nếu một ngôn ngữ không được hỗ trợ mặc định, người dùng có thể cài đặt thêm các tiện ích mở rộng (extensions) để bổ sung tính năng hỗ trợ cho ngôn ngữ đó.

*Giao diện trực quan, dễ sử dụng.*

VS Code có giao diện đơn giản nhưng rất trực quan, giúp lập trình viên dễ dàng làm quen và sử dụng ngay từ lần đầu tiên. Người dùng có thể tùy chỉnh giao diện theo sở thích cá nhân, bao gồm thay đổi theme, điều chỉnh phím tắt và cài đặt giao diện theo phong cách yêu thích. Bên cạnh đó, VS Code còn hỗ trợ chế độ Dark Mode, giúp giảm căng thẳng mắt khi làm việc trong thời gian dài.

*Tích hợp Git và hỗ trợ kiểm soát phiên bản.*

Một trong những điểm mạnh của VS Code là khả năng tích hợp Git ngay trong trình soạn thảo. Điều này giúp lập trình viên dễ dàng thực hiện các thao tác như commit, push, pull, merge code mà không cần mở phần mềm quản lý Git riêng biệt. Ngoài Git, VS Code còn hỗ trợ làm việc với các hệ thống quản lý mã nguồn khác như SVN, Mercurial thông qua các tiện ích mở rộng.

*IntelliSense – Gợi ý mã thông minh.*

VS Code sử dụng IntelliSense, một tính năng gợi ý mã thông minh giúp lập trình viên viết code nhanh hơn và chính xác hơn. IntelliSense có khả năng tự động hoàn thành mã, kiểm tra lỗi cú pháp và đưa ra gợi ý dựa trên ngữ cảnh của đoạn code đang viết. Điều này đặc biệt hữu ích khi làm việc với các ngôn ngữ có cú pháp phức tạp như JavaScript, Python hoặc C++.

*Debugging trực tiếp.*

VS Code hỗ trợ debug code trực tiếp, cho phép lập trình viên tìm và sửa lỗi mà không cần sử dụng thêm công cụ bên ngoài. Người dùng có thể đặt breakpoint, kiểm tra giá trị biến trong thời gian thực, theo dõi luồng chạy của chương trình và kiểm soát quá trình thực thi một cách dễ dàng. Điều này giúp giảm thời gian tìm lỗi và cải thiện hiệu suất làm việc.

*Tích hợp Terminal ngay trong quá trình soạn thảo.*

Một điểm tiện lợi của VS Code là khả năng tích hợp terminal ngay trong giao diện làm việc. Người dùng có thể mở terminal để chạy lệnh mà không cần chuyển đổi sang cửa sổ dòng lệnh bên ngoài. VS Code hỗ trợ nhiều loại shell như Command Prompt, PowerShell, Git Bash và Zsh, giúp lập trình viên làm việc linh hoạt với nhiều môi trường khác nhau.

*Hệ thống tiện ích mở rộng phong phú.*

VS Code có một kho tiện ích mở rộng khổng lồ trên Marketplace, giúp người dùng có thể bổ sung các tính năng mới theo nhu cầu. Một số tiện ích mở rộng phổ biến bao gồm Prettier (hỗ trợ định dạng mã nguồn tự động), ESLint (kiểm tra lỗi cú pháp JavaScript), Python Extension Pack (cung cấp công cụ hỗ trợ lập trình Python) và Live Server (hỗ trợ chạy file HTML trực tiếp trên trình duyệt mà không cần refresh thủ công). Nhờ vào hệ thống extension phong phú, VS Code có thể đáp ứng nhu cầu của nhiều đối tượng lập trình viên khác nhau.

### 2.1.3. Các tiện ích mở rộng – Extentions được sử dụng

*GitHub Copilot.*

GitHub Copilot là một trợ lý lập trình bằng AI, được phát triển bởi GitHub và OpenAI, giúp lập trình viên viết code nhanh hơn và hiệu quả hơn. Công cụ này hoạt động dưới dạng một extension (tiện ích mở rộng) cho các trình soạn thảo như Visual Studio Code (VS Code), Visual Studio, Neovim và JetBrains IDEs.

GitHub Copilot sử dụng mô hình AI dựa trên GPT-4 để phân tích ngữ cảnh của đoạn code hiện tại và đưa ra các gợi ý mã nguồn một cách thông minh. Nó có thể tự động hoàn thành một dòng code, một khối code hoặc thậm chí một hàm hoàn chỉnh. Điều này giúp tiết kiệm thời gian, giảm lỗi cú pháp và cải thiện hiệu suất lập trình.

GitHub Copilot hỗ trợ nhiều ngôn ngữ lập trình phổ biến như Python, JavaScript, TypeScript, Java, C++, C#, Go, Rust, Ruby, PHP, Swift, Kotlin, v.v. Điều này giúp nó phù hợp với nhiều loại dự án khác nhau, từ phát triển web đến lập trình hệ thống và trí tuệ nhân tạo.

*GitHub Copilot Chat.*

GitHub Copilot Chat là một tính năng nâng cao của GitHub Copilot, giúp lập trình viên tương tác với AI thông qua một chatbot ngay trong trình soạn thảo. Nó hoạt động giống như một trợ lý lập trình cá nhân, có thể trả lời câu hỏi, giải thích đoạn mã, đề xuất cách sửa lỗi và thậm chí hướng dẫn cách tối ưu hóa code.

GitHub Copilot Chat có thể giúp lập trình viên hiểu rõ hơn về một đoạn mã, giải thích cách hoạt động của một hàm hoặc hướng dẫn cách sử dụng một thư viện cụ thể. Nếu gặp lỗi trong chương trình, Copilot Chat có thể giúp phân tích lỗi, đề xuất cách khắc phục và thậm chí cung cấp một đoạn code sửa lỗi. Điều này giúp tiết kiệm thời gian debugging và giảm công sức tìm kiếm lỗi.

GitHub Copilot Chat còn có thể tự động tạo test case cho mã nguồn. Ví dụ, nếu em có một hàm trong Python, Copilot Chat có thể tự động tạo ra các unit test phù hợp bằng cách sử dụng framework như *unittest* hoặc *pytest*. Ngoài ra, khi tìm hiểu về công nghệ mới như React Native hay thư viện Tkinter của Python, Copilot Chat có thể đề xuất các tài nguyên hữu ích, bao gồm tài liệu chính thức, hướng dẫn thực hành và các bài viết liên quan.

*Python Extention*

Python Extension là tiện ích mở rộng quan trọng nhất dành cho lập trình viên Python trên Visual Studio Code. Đây là extension chính thức do Microsoft phát triển, cung cấp đầy đủ các công cụ giúp lập trình viên làm việc hiệu quả hơn với Python. Sau khi cài đặt extension, VS Code có thể tự động phát hiện các phiên bản Python đã cài trên hệ thống, bao gồm Python cài đặt trực tiếp, môi trường virtualenv, conda, hoặc WSL (Windows Subsystem for Linux) và cho phép chạy mã Python trực tiếp từ VS Code.

Ngoài ra, extension này còn tích hợp với Jupyter Notebook, giúp người dùng có thể chạy từng đoạn mã mà không cần sử dụng trình duyệt, điều này rất hữu ích cho dữ liệu khoa học, AI/ML, và tính toán số học. Đặc biệt, Python Extension hỗ trợ IntelliSense, cung cấp tính năng tự động hoàn thành mã thông minh, giúp lập trình viên viết code nhanh hơn và chính xác hơn. Đây là extension không thể thiếu nếu muốn làm việc với Python trên VS Code.

*Pylance*

Pylance là một tiện ích mở rộng mạnh mẽ giúp tăng cường tính năng IntelliSense cho Python trên VS Code. Extension này sử dụng công nghệ phân tích mã nguồn tiên tiến, giúp cung cấp gợi ý mã nhanh hơn, chính xác hơn và tối ưu hơn so với Python Extension thông thường.

Một trong những điểm nổi bật của Pylance là khả năng kiểm tra kiểu dữ liệu (Type Checking) theo chuẩn PEP 484, giúp lập trình viên phát hiện lỗi ngay từ khi viết code thay vì đợi đến khi chạy chương trình. Ngoài ra, Pylance còn cung cấp thông tin chi tiết về các hàm, lớp và biến khi di chuột qua, giúp lập trình viên dễ dàng hiểu hơn về mã nguồn mà họ đang làm việc.

Pylance giúp tăng tốc IntelliSense bằng cách nạp trước các thư viện và phân tích mã nhanh hơn, giúp trải nghiệm viết code mượt mà ngay cả trong các dự án lớn, mang lại trải nghiệm mượt mà và hiệu quả hơn khi lập trình với Python.

\*PEP 484 (Python Enhancement Proposal 484) là một đề xuất mở rộng cho Python nhằm giới thiệu hệ thống Type Hints (gợi ý kiểu dữ liệu) vào ngôn ngữ này. Nó được giới thiệu trong Python 3.5 và giúp lập trình viên có thể chú thích kiểu dữ liệu cho biến, tham số và giá trị trả về của hàm một cách rõ ràng hơn.

*Python Debugger*

Python Debugger, còn được gọi là *debugpy*, là một extension quan trọng giúp lập trình viên gỡ lỗi (debug) mã nguồn Python một cách hiệu quả hơn trên VS Code. Thay vì sử dụng phương pháp truyền thống như print() để kiểm tra giá trị biến, Python Debugger cho phép lập trình viên đặt breakpoint (điểm dừng) trên bất kỳ dòng nào trong mã nguồn. Khi chạy chương trình trong chế độ debug, mã nguồn sẽ tạm dừng tại breakpoint, cho phép lập trình viên kiểm tra giá trị của các biến và theo dõi luồng thực thi.

Python Debugger cũng hỗ trợ các tính năng Step Over, Step Into, Step Out, giúp lập trình viên điều hướng qua từng dòng code một cách linh hoạt. Ngoài ra, extension này còn tích hợp với Jupyter Notebook và hỗ trợ debug các ứng dụng web sử dụng Flask hoặc Django, giúp lập trình viên kiểm tra và khắc phục lỗi một cách dễ dàng hơn trong quá trình phát triển phần mềm.

*React Native Tools*

React Native Tools là một tiện ích mở rộng (extension) dành cho Visual Studio Code giúp lập trình viên phát triển ứng dụng React Native một cách dễ dàng và hiệu quả hơn. Đây là một extension chính thức do Microsoft phát triển, cung cấp nhiều công cụ hỗ trợ mạnh mẽ để viết, chạy và debug ứng dụng React Native trực tiếp từ VS Code mà không cần sử dụng dòng lệnh phức tạp.

React Native Tools cho phép lập trình viên khởi chạy ứng dụng React Native trên Android Emulator, iOS Simulator, hoặc thiết bị thực chỉ với một cú nhấp chuột. Thay vì phải chạy các lệnh như *npx react-native start* hoặc *npx react-native run-android*, em có thể thực hiện tất cả ngay trong VS Code thông qua giao diện tích hợp.

Extension này hỗ trợ debug trực tiếp với khả năng đặt breakpoint, kiểm tra biến và theo dõi luồng thực thi của ứng dụng. Điều này giúp lập trình viên phát hiện lỗi nhanh chóng mà không cần sử dụng Chrome DevTools. Ngoài ra, React Native Tools còn hỗ trợ Live Reload và Hot Reload, giúp cập nhật giao diện và logic ứng dụng ngay lập tức mà không cần khởi động lại.

Extension hỗ trợ cả React Native CLI truyền thống và Expo, giúp lập trình viên làm việc dễ dàng với cả hai môi trường. Em có thể mở dự án, kiểm tra lỗi và chạy ứng dụng trên nhiều thiết bị một cách linh hoạt.

React Native Tools cung cấp tính năng IntelliSense, giúp tự động hoàn thành mã, gợi ý API của React Native và phát hiện lỗi trong quá trình viết code. Điều này giúp lập trình viên viết code nhanh hơn, chính xác hơn và tránh các lỗi thường gặp khi lập trình React Native.

Em có thể dễ dàng xem log của ứng dụng React Native ngay trong VS Code mà không cần mở terminal riêng. Điều này giúp theo dõi các sự kiện trong ứng dụng, kiểm tra lỗi và tối ưu hóa hiệu suất.

React Native Tools là một extension không thể thiếu nếu đang phát triển ứng dụng React Native bằng VS Code. Nó giúp đơn giản hóa quá trình phát triển, debug nhanh hơn, chạy ứng dụng dễ dàng trên nhiều thiết bị và cải thiện trải nghiệm lập trình viên.

*ES7+ React/Redux/React-Native Snippets*

ES7+ React/Redux/React-Native Snippets là một tiện ích mở rộng mạnh mẽ dành cho Visual Studio Code, giúp lập trình viên tăng tốc quá trình viết mã cho React, Redux và React Native bằng cách cung cấp các đoạn mã (snippets) có sẵn. Thay vì phải viết thủ công các đoạn code lặp đi lặp lại, extension này cho phép tạo nhanh các thành phần React, Redux actions, reducers, hooks và nhiều đoạn code khác chỉ bằng một vài phím tắt, giúp tiết kiệm thời gian và giảm thiểu lỗi cú pháp.

Một trong những tính năng hữu ích nhất của extension này là khả năng tạo nhanh các React components. Chẳng hạn, có thể sử dụng phím tắt *rafce* để tạo một React Arrow Function Component, hoặc *rcc* để tạo một React Class Component chỉ trong tích tắc. Điều này giúp lập trình viên không cần phải nhớ toàn bộ cú pháp mà vẫn có thể viết code một cách hiệu quả. Ngoài ra, extension còn hỗ trợ đầy đủ React Hooks, giúp khai báo nhanh các hook phổ biến như *useState*, *useEffect*, *useContext* chỉ với một vài ký tự như *usf*, *uef*, *uct*.

Không chỉ dừng lại ở React, extension này còn hỗ trợ mạnh mẽ cho Redux và Redux Toolkit. Với các snippet như *rxaction*, *rxreducer*, *rxslice*, lập trình viên có thể nhanh chóng tạo các actions, reducers và slices, giúp quản lý trạng thái ứng dụng một cách dễ dàng hơn. Nếu đang làm việc với React Native, extension cũng cung cấp các snippet để tạo nhanh component React Native, giúp tiết kiệm thời gian khi xây dựng giao diện ứng dụng di động.

Ngoài ra, ES7+ React/Redux/React-Native Snippets còn hỗ trợ GraphQL với Apollo Client, giúp lập trình viên dễ dàng viết các truy vấn (queries) và mutations mà không cần nhớ cú pháp phức tạp. Với khả năng tùy chỉnh linh hoạt, có thể bật/tắt các snippet không cần thiết để giữ môi trường làm việc gọn gàng hơn.

ES7+ React/Redux/React-Native Snippets là một extension không thể thiếu đối với bất kỳ lập trình viên nào làm việc với React, Redux hoặc React Native. Nó giúp viết code nhanh hơn, tránh lỗi cú pháp, hỗ trợ nhiều thư viện phổ biến và tích hợp hoàn hảo với VS Code.

## 2.2. Ngôn ngữ lập trình Python

### 2.2.1. Giới thiệu

Python là một ngôn ngữ lập trình bậc cao, đa mục đích, thông dịch và hướng đối tượng, được phát triển bởi Guido van Rossum và ra mắt lần đầu tiên vào năm 1991. Python được thiết kế với triết lý giúp lập trình viên viết code một cách dễ đọc, dễ viết và dễ bảo trì, nhờ cú pháp đơn giản và rõ ràng. Python là một trong những ngôn ngữ phổ biến nhất thế giới, được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau như phát triển phần mềm, khoa học dữ liệu, trí tuệ nhân tạo (AI), học máy (Machine Learning), lập trình nhúng, phát triển web, tự động hóa, tài chính, an ninh mạng, và nhiều lĩnh vực khác.

Một trong những lý do chính khiến Python trở nên phổ biến là tính linh hoạt và dễ học. So với các ngôn ngữ khác như C++, Java hoặc JavaScript, Python có cú pháp trực quan hơn, giúp lập trình viên tập trung vào giải quyết vấn đề thay vì bị phân tâm bởi cú pháp phức tạp. Python cũng có một hệ sinh thái thư viện và framework phong phú, hỗ trợ nhiều lĩnh vực khác nhau, từ phân tích dữ liệu với Pandas, NumPy, đến phát triển web với Django, Flask, hoặc học máy với TensorFlow, PyTorch.

Python là một ngôn ngữ thông dịch, có nghĩa là chương trình Python được thực thi từng dòng một bởi trình thông dịch thay vì được biên dịch thành mã máy trước khi chạy. Điều này giúp Python linh hoạt hơn, dễ dàng kiểm tra và sửa lỗi, nhưng cũng có thể khiến tốc độ thực thi chậm hơn so với các ngôn ngữ biên dịch như C++ hoặc Java. Tuy nhiên, nhờ vào sự tối ưu hóa của các thư viện và công cụ hiện đại, Python vẫn có thể đạt hiệu suất cao trong nhiều ứng dụng quan trọng.

Ngoài ra, Python là một ngôn ngữ mã nguồn mở, có nghĩa là bất kỳ ai cũng có thể sử dụng, sửa đổi và phân phối nó miễn phí. Điều này đã giúp Python có một cộng đồng phát triển mạnh mẽ trên toàn thế giới, với hàng triệu lập trình viên đóng góp vào việc phát triển các thư viện, công cụ và tài nguyên học tập cho Python. Nhờ vào sự hỗ trợ mạnh mẽ từ cộng đồng, Python liên tục được cải tiến và cập nhật để đáp ứng nhu cầu ngày càng cao của ngành công nghệ.

Python là một ngôn ngữ lập trình mạnh mẽ, linh hoạt và dễ sử dụng, phù hợp cho cả lập trình viên mới bắt đầu và những chuyên gia có kinh nghiệm. Với khả năng ứng dụng rộng rãi, cộng đồng phát triển lớn mạnh và hệ sinh thái thư viện đa dạng, Python tiếp tục là một trong những lựa chọn hàng đầu cho các lập trình viên và doanh nghiệp trên toàn cầu.

### 2.2.2. Đặc điểm của ngôn ngữ

*Cú pháp đơn giản và dễ học*

Python được thiết kế với cú pháp gọn gàng, dễ đọc và dễ viết, giúp lập trình viên có thể nhanh chóng làm quen mà không cần phải học quá nhiều quy tắc phức tạp. Cú pháp của Python gần với ngôn ngữ tự nhiên, giúp mã nguồn dễ hiểu ngay cả đối với những người mới bắt đầu. Không giống như các ngôn ngữ như C++ hay Java, Python không yêu cầu dấu chấm phẩy (;) hoặc dấu ngoặc nhọn ({}) để xác định các khối lệnh, mà thay vào đó sử dụng thụt lề (indentation), giúp code trở nên rõ ràng và trực quan hơn. Điều này giúp giảm lỗi cú pháp và giúp việc bảo trì code dễ dàng hơn.

*Hỗ trợ lập trình đa mô hình*

Python là một ngôn ngữ lập trình đa mô hình, hỗ trợ nhiều phong cách lập trình khác nhau như lập trình hướng đối tượng (OOP), lập trình thủ tục (Procedural), lập trình hàm (Functional Programming). Điều này mang lại sự linh hoạt khi phát triển phần mềm, cho phép lập trình viên lựa chọn phong cách lập trình phù hợp nhất với yêu cầu của dự án. Với mô hình hướng đối tượng, Python giúp dễ dàng xây dựng và tái sử dụng các thành phần phần mềm, trong khi đó mô hình lập trình hàm giúp viết code ngắn gọn và hiệu quả hơn khi làm việc với dữ liệu.

*Thư viện và framework phong phú*

Một trong những lý do khiến Python trở nên phổ biến là hệ sinh thái thư viện vô cùng phong phú. Python có sẵn hàng ngàn thư viện và framework phục vụ cho nhiều lĩnh vực khác nhau. Ví dụ, trong khoa học dữ liệu, Python có các thư viện như NumPy, Pandas, Matplotlib giúp phân tích và trực quan hóa dữ liệu. Trong học máy và trí tuệ nhân tạo, các thư viện như TensorFlow, Keras, PyTorch giúp xây dựng các mô hình AI mạnh mẽ. Đối với phát triển web, Python có Django, Flask, FastAPI, giúp tạo ứng dụng web nhanh chóng và bảo mật. Với sự hỗ trợ từ cộng đồng lớn mạnh, số lượng thư viện của Python không ngừng tăng lên, giúp lập trình viên giải quyết nhiều bài toán phức tạp mà không cần viết lại từ đầu.

*Khả năng mở rộng và tích hợp mạnh mẽ*

Python có thể dễ dàng tích hợp với nhiều ngôn ngữ lập trình khác như C, C++, Java, JavaScript và thậm chí cả .NET. Điều này cho phép lập trình viên sử dụng Python để viết mã nhanh chóng và sau đó tối ưu hóa hiệu suất bằng cách sử dụng các thư viện viết bằng ngôn ngữ khác. Python cũng hỗ trợ tích hợp với các hệ quản trị cơ sở dữ liệu như MySQL, PostgreSQL, MongoDB và có thể làm việc với API RESTful hoặc GraphQL một cách dễ dàng. Điều này giúp Python trở thành một công cụ mạnh mẽ để phát triển ứng dụng phần mềm trong môi trường doanh nghiệp.

*Chạy trên nhiều nền tảng khác nhau*

Python là một ngôn ngữ đa nền tảng (cross-platform), có thể chạy trên Windows, macOS, Linux và thậm chí trên các thiết bị di động. Điều này có nghĩa là lập trình viên có thể viết code một lần và chạy nó trên nhiều hệ điều hành mà không cần thay đổi mã nguồn nhiều. Hơn nữa, Python có các công cụ như PyInstaller giúp đóng gói ứng dụng thành file thực thi (.exe hoặc .app), giúp việc phân phối phần mềm trở nên dễ dàng hơn.

*Cộng đồng lớn mạnh và hỗ trợ tuyệt vời*

Python có một cộng đồng phát triển rộng lớn với hàng triệu lập trình viên trên toàn thế giới. Đây là một trong những yếu tố quan trọng giúp Python phát triển mạnh mẽ và luôn cập nhật với những công nghệ mới nhất. Có rất nhiều tài liệu học tập, khóa học miễn phí, diễn đàn, blog, hội nhóm giúp lập trình viên dễ dàng tìm kiếm giải pháp khi gặp vấn đề. Một số nền tảng nổi bật như Stack Overflow, GitHub, Reddit, Real Python là nơi các lập trình viên có thể trao đổi, học hỏi và phát triển kỹ năng Python của mình.

*Tính linh hoạt và khả năng ứng dụng rộng rãi*

Python không bị giới hạn trong một lĩnh vực cụ thể mà có thể được sử dụng trong nhiều ngành công nghiệp khác nhau. Từ phát triển web, khoa học dữ liệu, trí tuệ nhân tạo, đến lập trình nhúng, tự động hóa, tài chính và bảo mật mạng, Python có thể đáp ứng được mọi nhu cầu lập trình. Khả năng linh hoạt này giúp Python trở thành lựa chọn hàng đầu cho cả cá nhân học lập trình và doanh nghiệp đang tìm kiếm một công cụ phát triển phần mềm mạnh mẽ.

### 2.2.3. Ứng dụng của Python

*Khoa học dữ liệu và phân tích dữ liệu*

Python là một trong những ngôn ngữ phổ biến nhất trong lĩnh vực khoa học dữ liệu và phân tích dữ liệu, nhờ vào các thư viện mạnh mẽ như NumPy, Pandas, Matplotlib, Seaborn. Với các công cụ này, lập trình viên có thể dễ dàng xử lý, thao tác và trực quan hóa dữ liệu để tìm ra các xu hướng, mô hình và thông tin hữu ích. Python cũng được sử dụng trong Big Data, giúp phân tích và xử lý khối lượng dữ liệu khổng lồ trong các hệ thống như Hadoop và Spark. Các doanh nghiệp trong lĩnh vực tài chính, y tế, thương mại điện tử đều sử dụng Python để phân tích dữ liệu và đưa ra quyết định chiến lược dựa trên dữ liệu.

*Lập trình nhúng và Internet of Things (IoT)*

Python cũng được sử dụng rộng rãi trong lập trình nhúng và phát triển các hệ thống IoT (Internet of Things). Các thiết bị như Raspberry Pi, Arduino có thể chạy Python để điều khiển cảm biến, thu thập dữ liệu và thực hiện các tác vụ tự động. Python giúp lập trình viên dễ dàng tạo ra các ứng dụng IoT để theo dõi môi trường, điều khiển thiết bị thông minh trong nhà và phát triển robot.

*Ứng dụng trong tài chính và giao dịch tự động*

Python được sử dụng rộng rãi trong ngành tài chính, ngân hàng và giao dịch chứng khoán tự động (Algorithmic Trading). Các thư viện như QuantLib, zipline, backtrader giúp lập trình viên phân tích dữ liệu tài chính, tối ưu hóa danh mục đầu tư và phát triển các thuật toán giao dịch tự động. Python cũng được sử dụng để xây dựng các hệ thống quản lý rủi ro, dự báo xu hướng thị trường và tự động hóa báo cáo tài chính.

*Trí tuệ nhân tạo (AI) và học máy (Machine Learning)*

Python đóng vai trò quan trọng trong lĩnh vực trí tuệ nhân tạo (AI) và học máy (Machine Learning) nhờ vào các thư viện nổi tiếng như TensorFlow, PyTorch, Scikit-learn, Keras. Những thư viện này cung cấp các công cụ mạnh mẽ để xây dựng, huấn luyện và triển khai các mô hình AI từ cơ bản đến nâng cao, bao gồm nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên (NLP), chatbot, hệ thống gợi ý và xe tự hành. Các công ty công nghệ lớn như Google, Facebook, Amazon và Tesla đều sử dụng Python để phát triển các ứng dụng AI tiên tiến.

*Phát triển web và ứng dụng*

Python là một lựa chọn phổ biến trong phát triển web nhờ vào các framework mạnh mẽ như Django, Flask, FastAPI. Với những framework này, lập trình viên có thể xây dựng các ứng dụng web nhanh chóng, bảo mật và có khả năng mở rộng cao. Django là một framework mạnh mẽ giúp phát triển các trang web động phức tạp như Instagram hoặc Pinterest, trong khi Flask và FastAPI được sử dụng để xây dựng các dịch vụ web gọn nhẹ, API RESTful và ứng dụng microservices. Python cũng có thể được sử dụng để phát triển backend cho các ứng dụng web, kết nối với cơ sở dữ liệu như PostgreSQL, MySQL, MongoDB.

*Tự động hóa và DevOps*

Python là một công cụ mạnh mẽ để tự động hóa các tác vụ thủ công nhờ vào các thư viện như Selenium, BeautifulSoup, Requests, Paramiko. Lập trình viên có thể sử dụng Python để tự động hóa các công việc như quản lý tệp tin, gửi email hàng loạt, thu thập dữ liệu từ website (web scraping) hoặc kiểm tra hệ thống. Trong lĩnh vực DevOps, Python cũng được sử dụng để viết script giúp quản lý server, triển khai ứng dụng và giám sát hệ thống. Các công cụ như Ansible, Fabric, SaltStack dựa trên Python giúp quản trị viên hệ thống tự động hóa quy trình quản lý hạ tầng CNTT một cách hiệu quả.

*Phát triển trò chơi (Game Development)*

Python cũng được sử dụng trong lập trình game, đặc biệt với thư viện Pygame, giúp lập trình viên tạo ra các trò chơi 2D một cách dễ dàng. Ngoài ra, Python còn được sử dụng trong các công cụ phát triển game như Blender Game Engine, Panda3D để tạo các mô hình 3D và môi trường game. Nhiều công ty game sử dụng Python để xây dựng các công cụ hỗ trợ phát triển, chẳng hạn như Unity sử dụng Python để tạo script giúp phát triển game nhanh hơn.

*An ninh mạng và kiểm thử bảo mật*

Python là một trong những ngôn ngữ hàng đầu trong lĩnh vực an ninh mạng và kiểm thử bảo mật. Các chuyên gia bảo mật sử dụng Python để viết script kiểm thử hệ thống, phát hiện lỗ hổng và khai thác bảo mật. Các công cụ như Metasploit, Scapy, Nmap, Requests được viết bằng Python giúp phân tích lưu lượng mạng, kiểm tra lỗ hổng và thực hiện các cuộc tấn công giả lập để bảo vệ hệ thống. Python cũng được sử dụng trong lĩnh vực mật mã học (cryptography) để mã hóa và bảo vệ dữ liệu.

*Ứng dụng trong giáo dục và nghiên cứu*

Python là một ngôn ngữ lý tưởng để giảng dạy lập trình trong các trường học và đại học nhờ vào cú pháp đơn giản, dễ đọc và dễ học. Nhiều khóa học lập trình cơ bản trên thế giới sử dụng Python để giúp sinh viên làm quen với các khái niệm lập trình như biến, vòng lặp, hàm và hướng đối tượng. Ngoài ra, Python cũng được sử dụng trong các nghiên cứu khoa học nhờ vào khả năng tính toán mạnh mẽ và thư viện hỗ trợ đa dạng như SciPy, SymPy trong toán học và vật lý.

*Ứng dụng trong blockchain và tiền điện tử*

Python cũng đóng một vai trò quan trọng trong lĩnh vực blockchain và tiền điện tử. Các nhà phát triển sử dụng Python để xây dựng các ứng dụng blockchain, tạo hợp đồng thông minh và phân tích dữ liệu chuỗi khối (blockchain analytics). Các thư viện như Web3.py, PyCryptodome giúp lập trình viên dễ dàng kết nối với các nền tảng blockchain như Ethereum, Binance Smart Chain.

### 2.2.4. Các thư viện Python được sử dụng

#### 2.2.4.1. Thư viện Tkinter

Tkinter là thư viện giao diện đồ họa mặc định của Python, giúp lập trình viên dễ dàng tạo các ứng dụng GUI (Graphical User Interface) mà không cần cài đặt thêm bất kỳ thư viện bên ngoài nào. Tkinter được xây dựng dựa trên Tk GUI toolkit, một công cụ mạnh mẽ hỗ trợ phát triển giao diện đồ họa trên nhiều nền tảng như Windows, macOS và Linux. Vì được tích hợp sẵn trong Python, Tkinter là lựa chọn phổ biến nhất để tạo giao diện đồ họa mà không cần cài đặt thêm thư viện bên ngoài.

##### 2.2.4.1.1. Đặc điểm của thư viện Tkinter

*Dễ học và dễ sử dụng*: Tkinter có cú pháp đơn giản, dễ hiểu, phù hợp cho cả người mới học lập trình và lập trình viên chuyên nghiệp.

*Tích hợp sẵn trong Python*: Không cần cài đặt bổ sung, có thể sử dụng ngay bằng cách import tkinter.

*Đa nền tảng*: Chạy được trên Windows, macOS và Linux mà không cần thay đổi mã nguồn.

*Cung cấp nhiều widget đa dạng*: Tkinter hỗ trợ nhiều thành phần giao diện như nút bấm (Button), nhãn (Label), ô nhập (Entry), hộp danh sách (Listbox), thanh cuộn (Scrollbar), menu (Menu), khung vẽ (Canvas),... giúp xây dựng giao diện phong phú.

*Hỗ trợ lập trình hướng sự kiện (Event-Driven)*: Ứng dụng có thể phản hồi lại các sự kiện từ người dùng như nhấp chuột, nhập liệu, nhấn phím,...

*Có thể tùy chỉnh giao diện*: Hỗ trợ thay đổi màu sắc, font chữ, kích thước và kiểu dáng của các thành phần giao diện.

##### 2.2.4.1.2. Thành phần chính trong thư viện Tkinter

Tkinter cung cấp nhiều thành phần giao diện (widget) giúp xây dựng ứng dụng một cách linh hoạt:

*Label*: Hiển thị văn bản hoặc hình ảnh.

*Button*: Tạo nút bấm để thực hiện hành động khi được nhấn.

*Entry*: Ô nhập văn bản cho người dùng nhập dữ liệu.

*Text*: Ô nhập văn bản đa dòng.

*Frame*: Khung chứa các widget khác, giúp sắp xếp bố cục giao diện.

*Canvas*: Khung vẽ để vẽ đồ họa, hình ảnh hoặc biểu đồ.

*Listbox*: Hộp danh sách hiển thị nhiều lựa chọn.

*Scrollbar*: Thanh cuộn để điều hướng nội dung dài.

*Menu*: Thanh menu giúp tạo các tùy chọn điều khiển ứng dụng.

##### 2.2.4.1.3. Ứng dụng trong hệ thống ICarP

#### 2.2.4.2. Thư viện OpenCV

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở chuyên dùng để xử lý hình ảnh và thị giác máy tính (Computer Vision). Được phát triển bởi Intel vào năm 2000, OpenCV đã trở thành một trong những thư viện phổ biến nhất trong lĩnh vực xử lý ảnh, nhận diện khuôn mặt, thị giác máy và trí tuệ nhân tạo. Thư viện này hỗ trợ nhiều ngôn ngữ lập trình, bao gồm Python, C++, Java, và có thể chạy trên nhiều nền tảng như Windows, macOS, Linux, Android và iOS.

##### 2.2.4.2.1. Đặc điểm của thư viện OpenCV

*Mã nguồn mở và miễn phí*: OpenCV là một thư viện mã nguồn mở, giúp lập trình viên có thể sử dụng và tùy chỉnh theo nhu cầu mà không tốn phí.

*Hiệu suất cao*: Được tối ưu hóa bằng ngôn ngữ C/C++, OpenCV có khả năng xử lý hình ảnh nhanh hơn so với các thư viện khác.

*Hỗ trợ nhiều thuật toán xử lý ảnh và thị giác máy*: Bao gồm nhận diện khuôn mặt, phát hiện đối tượng, phân đoạn ảnh, theo dõi chuyển động, xử lý video, v.v.

*Khả năng làm việc với trí tuệ nhân tạo (AI) và học máy*: OpenCV tích hợp với các framework AI như TensorFlow, PyTorch và scikit-learn để xử lý hình ảnh thông minh hơn.

*Đa nền tảng*: Hỗ trợ chạy trên Windows, macOS, Linux, Raspberry Pi, và các thiết bị di động.

*Hỗ trợ tăng tốc GPU*: OpenCV có thể sử dụng CUDA và OpenCL để tăng tốc xử lý trên GPU, giúp cải thiện hiệu suất trong các tác vụ nặng.

##### 2.2.4.2.2. Ứng dụng của thư viện OpenCV

*Nhận diện khuôn mặt*: Ứng dụng trong bảo mật, mở khóa bằng khuôn mặt, chấm công bằng nhận diện.

*Xử lý ảnh y tế*: Giúp phân tích ảnh chụp X-ray, MRI, CT scan để hỗ trợ bác sĩ chẩn đoán bệnh.

*Thị giác máy trong công nghiệp*: Kiểm tra chất lượng sản phẩm trong dây chuyền sản xuất.

*Phát hiện và theo dõi đối tượng*: Ứng dụng trong xe tự hành, camera giám sát thông minh, theo dõi chuyển động.

*Xử lý video thời gian thực*: Sử dụng trong các hệ thống giám sát, livestream với bộ lọc ảnh.

*Thực tế ảo (AR) và thực tế tăng cường (VR)*: Tích hợp vào các ứng dụng AR như Snapchat, Instagram để tạo hiệu ứng khuôn mặt.

##### 2.2.4.2.3. Ứng dụng trong hệ thống ICarP

#### 2.2.4.3. Thư viện EasyOCR

EasyOCR là một thư viện mã nguồn mở mạnh mẽ dùng để nhận dạng ký tự quang học (OCR - Optical Character Recognition) trong hình ảnh. Thư viện này được phát triển bởi Jaided AI, giúp trích xuất văn bản từ hình ảnh hoặc tài liệu một cách nhanh chóng và chính xác. EasyOCR được xây dựng dựa trên PyTorch, sử dụng các mô hình học sâu (Deep Learning) để nhận dạng chữ viết từ hơn 80 ngôn ngữ, bao gồm cả tiếng Việt.

##### 2.2.4.3.1. Đặc điểm của thư viện EasyOCR

*Dễ cài đặt và sử dụng*: Cung cấp giao diện đơn giản, chỉ cần vài dòng lệnh để nhận dạng văn bản từ hình ảnh.

*Hỗ trợ hơn 80 ngôn ngữ*: Nhận dạng tốt nhiều ngôn ngữ, kể cả các ngôn ngữ có ký tự đặc biệt như Trung Quốc, Nhật Bản, Hàn Quốc và tiếng Việt.

*Dựa trên Deep Learning*: Sử dụng mạng nơ-ron sâu để cải thiện độ chính xác trong nhận dạng ký tự.

*Hỗ trợ nhiều định dạng ảnh*: Có thể xử lý các định dạng ảnh phổ biến như PNG, JPG, BMP, TIFF.

*Nhanh và hiệu quả*: Tối ưu hóa hiệu suất để có thể hoạt động tốt ngay cả trên máy tính cá nhân mà không cần GPU mạnh.

*Có thể tùy chỉnh và mở rộng*: Người dùng có thể huấn luyện lại mô hình để cải thiện độ chính xác đối với các trường hợp đặc biệt.

##### 2.2.4.3.2. Ứng dụng của thư viện EasyOCR

*Nhận dạng văn bản trong tài liệu số hóa*: Trích xuất văn bản từ sách, hóa đơn, CMND, bằng lái xe, hợp đồng, v.v.

*Tự động hóa nhập dữ liệu*: Giúp nhận diện và nhập liệu tự động từ giấy tờ, biểu mẫu.

*Nhận diện biển số xe*: Dùng trong các hệ thống giám sát giao thông hoặc kiểm soát xe vào/ra bãi đỗ.

*Dịch văn bản từ hình ảnh*: Kết hợp với Google Translate để dịch văn bản trên ảnh.

*Nhận diện chữ viết tay*: Hỗ trợ nhận dạng chữ viết tay trong một số trường hợp cụ thể.

##### 2.2.4.3.3. Ứng dụng trong hệ thống ICarP

#### 2.2.4.4. Thư viện Numpy

NumPy (Numerical Python) là một thư viện mạnh mẽ của Python, chuyên dùng để tính toán khoa học, xử lý dữ liệu số và làm việc với mảng đa chiều (array). Đây là thư viện cốt lõi trong hệ sinh thái khoa học dữ liệu của Python, hỗ trợ tối ưu hóa hiệu suất khi làm việc với số lượng lớn dữ liệu.

NumPy cung cấp một kiểu dữ liệu mảng ndarray (N-dimensional array), có khả năng lưu trữ và xử lý dữ liệu hiệu quả hơn so với danh sách (list) thông thường của Python. Thư viện này còn hỗ trợ các phép toán đại số tuyến tính, ma trận, thống kê và các hàm toán học nhanh chóng, giúp tiết kiệm thời gian và tài nguyên khi làm việc với dữ liệu lớn.

##### 2.2.4.4.1. Đặc điểm của thư viện Numpy

*Xử lý mảng hiệu suất cao*: NumPy cho phép thao tác với mảng lớn nhanh hơn so với danh sách Python thông thường.

*Hỗ trợ tính toán ma trận và đại số tuyến tính*: Cung cấp nhiều hàm hỗ trợ cho các bài toán liên quan đến ma trận, vectơ.

*Tiêu tốn ít bộ nhớ hơn so với danh sách Python (Python List)*: Nhờ vào kiểu dữ liệu cố định và tối ưu hóa bộ nhớ.

*Dễ dàng tích hợp với các thư viện khác*: NumPy là nền tảng của nhiều thư viện trong khoa học dữ liệu như Pandas, SciPy, Matplotlib, TensorFlow.

*Hỗ trợ tính toán song song với GPU*: Có thể kết hợp với CUDA để chạy trên GPU, giúp tăng tốc độ xử lý.

##### 2.2.4.4.2. Ứng dụng trọng hệ thống ICarP

#### 2.2.4.5. Thư viện RE

Thư viện *re* là một thư viện tích hợp sẵn trong Python, được sử dụng để xử lý biểu thức chính quy (Regular Expressions - RegEx). Biểu thức chính quy là một công cụ mạnh mẽ để tìm kiếm, thay thế, kiểm tra và xử lý chuỗi ký tự theo mẫu nhất định.

Thư viện *re* giúp lập trình viên thao tác với chuỗi một cách linh hoạt, đặc biệt hữu ích khi xử lý dữ liệu văn bản, phân tích log, kiểm tra định dạng email, số điện thoại, tách dữ liệu từ văn bản, v.v.

##### 2.2.4.5.1. Biểu thức chính quy - Regular Expressions

|  |  |  |
| --- | --- | --- |
| Ký hiệu | Ý nghĩa | Ví dụ |
| . | Bất kỳ ký tự nào trừ xuống dòng | "c.t" khớp với "cat", "cut" |
| ^ | Bắt đầu chuỗi | "^Hello" khớp với "Hello world", nhưng không khớp với "Say Hello" |
| $ | Kết thúc chuỗi | "world$" khớp với "Hello world", nhưng không khớp với "world Hello" |
| \* | Lặp lại 0 hoặc nhiều lần | "ab\*" khớp với "a", "ab", "abb", "abbb" |
| + | Lặp lại 1 hoặc nhiều lần | "ab+" khớp với "ab", "abb", nhưng không khớp với "a" |
| ? | Lặp lại 0 hoặc 1 lần | "ab?" khớp với "a", "ab" |
| {n} | Lặp lại đúng n lần | "a{3}" khớp với "aaa" nhưng không khớp với "aa" |
| {n,} | Lặp lại ít nhất n lần | "a{2,}" khớp với "aa", "aaa", "aaaa" |
| {n,m} | Lặp lại từ n đến m lần | "a{2,4}" khớp với "aa", "aaa", "aaaa" nhưng không khớp với "a" |
| [] | Ký tự trong khoảng | "[a-z]" khớp với bất kỳ chữ thường nào từ a đến z |
| \ | Dùng để thoát ký tự đặc biệt | "3\+3=6" khớp với "3+3=6" |

*Cú pháp cơ bản của biểu thức chính quy*

|  |  |  |
| --- | --- | --- |
| Ký hiệu | Ý nghĩa | Ví dụ |
| \d | Bất kỳ chữ số nào (0-9) | "\d{3}" khớp với "123" nhưng không khớp với "abc" |
| \D | Bất kỳ ký tự nào không phải số | "\D+" khớp với "abc", "hello" nhưng không khớp với "123" |
| \w | Ký tự chữ và số (A-Z, a-z, 0-9, \_) | "\w+" khớp với "hello123", "Python\_3" |
| \W | Ký tự không phải chữ hoặc số | "\W+" khớp với "@!$" |
| \s | Khoảng trắng, tab hoặc xuống dòng | "Hello\sWorld" khớp với "Hello World" |
| \S | Bất kỳ ký tự nào không phải khoảng trắng | "\S+" khớp với "Hello", "123" nhưng không khớp với " " |
| \b | Biên từ (word boundary) | "\bPython\b" chỉ khớp với "Python", không khớp với "Python3" |
| \B | Không phải biên từ | "\Bcat\B" không khớp với " cat ", nhưng có thể khớp với "catalog" |

*Ký tự đặc biệt của biểu thức chính quy*

##### 2.2.4.5.2. Ứng dụng của thư viện RE

*Tìm kiếm và trích xuất dữ liệu từ chuỗi*: Dễ dàng tìm các mẫu ký tự trong đoạn văn bản lớn.

*Kiểm tra định dạng chuỗi*: Xác minh số điện thoại, email, địa chỉ IP, mã bưu điện, .v.v.

*Thay thế nội dung theo mẫu*: Thay thế nhanh các chuỗi con trong văn bản dựa trên mẫu xác định.

*Tách chuỗi theo mẫu cụ thể*: Hữu ích khi cần phân tách dữ liệu có định dạng đặc biệt.

2.2.4.5.3. Ứng dụng trong hệ thống ICarP

## 2.3. You Only Live Once – YOLO

You Only Look Once (YOLO) là một thuật toán phát hiện đối tượng thời gian thực tiên tiến, được giới thiệu vào năm 2015 bởi Joseph Redmon, Santosh Divvala, Ross Girshick và Ali Farhadi trong bài báo nghiên cứu nổi tiếng You Only Look Once: Unified, Real-Time Object Detection.

Các tác giả đã tiếp cận vấn đề phát hiện đối tượng như một bài toán hồi quy thay vì phân loại, bằng cách chia không gian ảnh thành các hộp giới hạn (bounding boxes) và gán xác suất cho từng đối tượng được phát hiện, sử dụng một mạng nơ-ron tích chập (CNN) duy nhất.

### 2.3.1. CNN – Convolutional Neural Network

#### 2.3.1.1. Giới thiệu chung

CNN (Convolutional Neural Network) là một loại mạng nơ-ron nhân tạo (Artificial Neural Network - ANN) được thiết kế để xử lý dữ liệu hình ảnh và chuỗi không gian. CNN được sử dụng rộng rãi trong thị giác máy tính (Computer Vision) như nhận diện hình ảnh, phân loại đối tượng, phát hiện vật thể (YOLO, Faster R-CNN), phân đoạn ảnh và nhiều ứng dụng khác.

Mạng Nơ-ron Tích Chập (CNNs) đã cách mạng hóa lĩnh vực thị giác máy tính bằng cách giúp máy móc đạt được độ chính xác chưa từng có trong các nhiệm vụ như phân loại ảnh, nhận diện đối tượng và phân đoạn ảnh. CNNs, bắt nguồn từ công trình của Yann LeCun vào cuối những năm 1980, được lấy cảm hứng từ hệ thống thị giác của con người và xử lý dữ liệu hình ảnh bằng cách sử dụng cấu trúc phân cấp.

Mạng nơ-ron tích chập (CNNs) là các mạng nơ-ron sâu được thiết kế để xử lý dữ liệu dạng lưới, chẳng hạn như hình ảnh. Không giống như các mạng nơ-ron truyền thống, CNN sử dụng các lớp tích chập để tự động học các hệ thống đặc trưng không gian theo thứ bậc. Chúng bao gồm nhiều lớp, mỗi lớp thực hiện một chức năng cụ thể trong việc xử lý và biến đổi dữ liệu đầu vào để trích xuất các mẫu có ý nghĩa.

#### 2.3.1.2. Đặc điểm chính của CNNs

*Trường tiếp nhận cục bộ (Local Receptive Fields)*: CNNs sử dụng các bộ lọc nhỏ (filter/kernel) di chuyển trên ảnh đầu vào, tập trung vào các vùng cụ thể để phát hiện các đặc trưng cục bộ như cạnh, kết cấu và mẫu hình.

*Chia sẻ trọng số (Weight Sharing)*: Cùng một bộ lọc được áp dụng cho nhiều vùng khác nhau của ảnh đầu vào, giúp giảm số lượng tham số và độ phức tạp tính toán, đồng thời cho phép mạng nhận diện đối tượng bất kể vị trí của chúng.

*Lớp gộp (Pooling)*: Lớp pooling giúp giảm kích thước không gian của bản đồ đặc trưng, làm cho mạng hiệu quả hơn và thích ứng với các thay đổi trong hình ảnh. Max pooling, chẳng hạn, chọn giá trị lớn nhất trong mỗi vùng cửa sổ, giúp giữ lại thông tin quan trọng trong khi giảm số lượng tham số.

Cấu trúc của CNN kết hợp các yếu tố như trường tiếp nhận cục bộ, chia sẻ trọng số và pooling, giúp chúng trở nên cực kỳ hiệu quả và đáng tin cậy trong xử lý hình ảnh. Điều này giúp CNN có thể thực hiện các nhiệm vụ như phân loại hình ảnh, nhận diện đối tượng và phân đoạn ảnh, khiến chúng trở thành công cụ không thể thiếu trong lĩnh vực thị giác máy tính.

#### 2.3.1.3. Cấu trúc của CNNs

*Lớp tích chập (Convolutional Layer):* Đây là thành phần quan trọng nhất của CNN. Nó hoạt động bằng cách sử dụng một bộ lọc (kernel) trượt trên ảnh đầu vào và tính tích vô hướng giữa bộ lọc và vùng tiếp nhận cục bộ của ảnh. Phép toán này giúp phát hiện các đặc trưng như cạnh, kết cấu và mẫu hình.

*Lớp gộp (Pooling Layer):* Lớp này có nhiệm vụ giảm kích thước không gian của bản đồ đặc trưng (feature map), giúp giảm tải tính toán và tăng độ bền vững của đặc trưng.

*Lớp kết nối đầy đủ (Fully Connected Layer):* Mỗi nơ-ron trong lớp này được kết nối với tất cả các nơ-ron trong lớp trước đó, giúp tích hợp các đặc trưng đã trích xuất để tạo ra dự đoán cuối cùng.

*Dropout:* Trong quá trình huấn luyện, Dropout sẽ vô hiệu hóa ngẫu nhiên một phần các nơ-ron để tránh hiện tượng overfitting và giúp mô hình tổng quát hóa tốt hơn.

*Hàm kích hoạt (Activation Function):* Các hàm kích hoạt đưa tính phi tuyến vào mạng, giúp mô hình có thể học được các mẫu phức tạp. Các hàm phổ biến bao gồm ReLU và Sigmoid.

#### 2.3.1.4. Nguyên lý hoạt động của CNNs

##### 2.3.1.4.1. Phép toán tích chập - Convolution Operation.

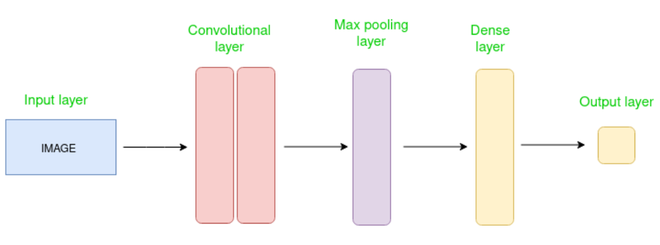
Phép toán tích chập hoạt động bằng cách trượt một bộ lọc trên ảnh đầu vào và tính tích vô hướng giữa bộ lọc và vùng ảnh cục bộ. Phép toán này tạo ra một bản đồ đặc trưng (feature map) giúp làm nổi bật các đặc trưng được phát hiện.

Công thức tích chập:

*I* là ảnh đầu vào, *K* là bộ lọc (kernel), *(i, j)* là tọa độ của bản đồ đặc trưng đầu ra.

##### 2.3.1.4.2. Quy trình hoạt động của CNNs

Mạng Nơ-ron Tích Chập (CNNs) được thiết kế để xử lý và phân tích dữ liệu hình ảnh bằng cách tự động và thích ứng học các đặc trưng không gian theo thứ bậc.



*Cấu trúc của CNNs*

*Lớp đầu vào (Input Layer)*

Lớp đầu vào của CNN nhận giá trị điểm ảnh thô từ hình ảnh. Một hình ảnh màu thường có ba kênh (RGB), trong khi một hình ảnh xám chỉ có một kênh. Ví dụ, một hình ảnh màu có kích thước 32x32 pixel sẽ có đầu vào với kích thước 32x32x3.

*Lớp tích chập (Convolutional Layers)*

Hoạt động chính trong lớp tích chập là phép tích chập (convolution), trong đó các bộ lọc (filters hay kernels) được áp dụng lên dữ liệu đầu vào. Một bộ lọc là một ma trận nhỏ (ví dụ: 3x3) di chuyển trên hình ảnh đầu vào và thực hiện phép nhân từng phần tử với tổng hợp để tạo ra một giá trị đầu ra duy nhất. Kết quả của phép tích chập là một bản đồ đặc trưng (feature map) hoặc bản đồ kích hoạt (activation map).

*Bước nhảy và đệm (Stride and Padding)*

Bước nhảy (Stride): Xác định khoảng cách bộ lọc di chuyển trên hình ảnh đầu vào. Bước nhảy là 1 nghĩa là bộ lọc di chuyển từng pixel một, trong khi bước nhảy là 2 nghĩa là bộ lọc di chuyển hai pixel một lần. Bước nhảy lớn hơn tạo ra bản đồ đặc trưng nhỏ hơn.

Đệm (Padding): Được sử dụng để kiểm soát kích thước không gian của bản đồ đặc trưng đầu ra.

* "Same" padding: Thêm các số 0 xung quanh biên của đầu vào, giúp giữ kích thước bản đồ đặc trưng đầu ra bằng với đầu vào.
* "Valid" padding: Không thêm bất kỳ số 0 nào, làm cho bản đồ đặc trưng đầu ra nhỏ hơn so với đầu vào.

*Hàm kích hoạt (Activation Functions)*

Hàm kích hoạt rất quan trọng trong mạng nơ-ron vì chúng đưa vào tính phi tuyến, cho phép mạng học các mô hình phức tạp. Nếu không có các hàm kích hoạt phi tuyến, mạng nơ-ron sẽ hoạt động như một mô hình tuyến tính, bất kể có bao nhiêu lớp.

*Hàm ReLU (Rectified Linear Unit)*: Hàm kích hoạt ReLU được sử dụng rộng rãi trong CNN vì tính đơn giản và hiệu quả của nó.

Điều này có nghĩa là nếu giá trị đầu vào dương, ReLU trả về chính giá trị đó; nếu âm, nó trả về 0. ReLU giúp giảm bớt vấn đề *biến mất gradient (vanishing gradient problem)*, giúp quá trình huấn luyện nhanh hơn và hiệu quả hơn bằng cách duy trì gradient không bị mất đi đối với các giá trị dương.

*Hàm Sigmoid*: Hàm Sigmoid thường được sử dụng trong các bài toán phân loại nhị phân, vì nó chuyển đổi bất kỳ giá trị đầu vào nào thành giá trị trong khoảng từ 0 đến 1, giúp biểu diễn xác suất.

Mặc dù hữu ích trong một số ứng dụng, hàm Sigmoid có thể gặp vấn đề với *biến mất gradient (vanishing gradient problem)* và tốc độ hội tụ chậm hơn so với ReLU.

*Lớp Pooling (Pooling Layers)*

Các lớp Pooling được sử dụng để giảm kích thước không gian của bản đồ đặc trưng, giúp giảm tải tính toán và số lượng tham số của mạng. Pooling thường được sử dụng sau lớp tích chập và lớp kích hoạt.

* Max Pooling: Lấy giá trị lớn nhất từ mỗi vùng nhỏ (ví dụ: 2x2) trong bản đồ đặc trưng.
* Average Pooling: Tính giá trị trung bình của từng vùng nhỏ.

*Xếp chồng các lớp (Stacking Layers)*

CNNs được tạo thành từ nhiều lớp tích chập và pooling được xếp chồng lên nhau.

* Các lớp đầu tiên phát hiện các đặc trưng cơ bản như cạnh và kết cấu.
* Các lớp sau phát hiện các đặc trưng phức tạp hơn như cấu trúc và vật thể.

Cấu trúc này giúp CNNs có thể học các đặc trưng ở nhiều mức độ khác nhau, làm cho chúng cực kỳ hiệu quả trong nhận diện hình ảnh.

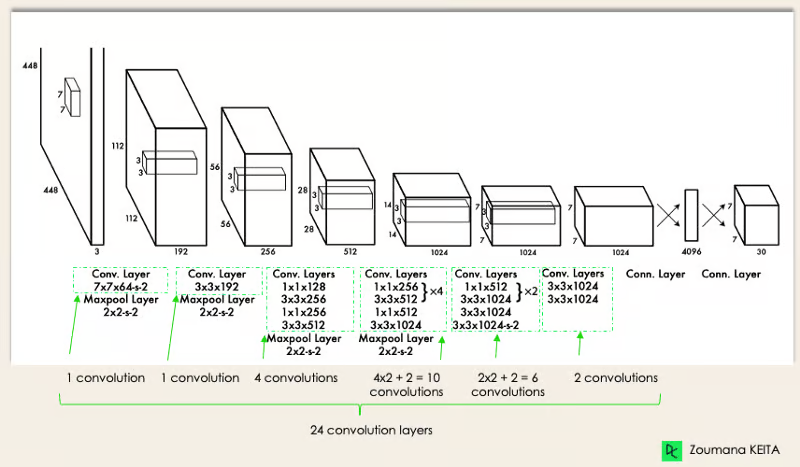
*Lớp kết nối đầy đủ (Fully Connected Layers)*

Sau nhiều lớp tích chập và pooling, các lớp fully connected (FC) đảm nhiệm phần suy luận cao cấp của mạng. Mỗi neuron trong lớp kết nối với tất cả neuron ở lớp trước. Các đặc trưng đã được trích xuất từ các lớp trước đó sẽ được kết hợp để đưa ra dự đoán cuối cùng.

*Lớp Dropout (Dropout Layer)*

Dropout được sử dụng để ngăn chặn quá khớp (overfitting). Trong quá trình huấn luyện, Dropout sẽ ngẫu nhiên đặt một phần số neuron thành 0 tại mỗi lần cập nhật, giúp mô hình không quá phụ thuộc vào một số neuron cụ thể, tăng khả năng tổng quát hóa của mô hình.

### 2.3.2. Cấu trúc của YOLO



*Cấu trúc của mô hình YOLOv1*

Mô hình YOLOv1 bao gồm 24 lớp tích chập (convolutional layers), 4 lớp pooling cực đại (max-pooling layers), tiếp theo là 2 lớp kết nối đầy đủ (fully connected layers). Mô hình hoạt động theo trình tự sau:

- Thay đổi kích thước hình ảnh đầu vào thành 448×448 trước khi đưa vào mạng tích chập (CNN).

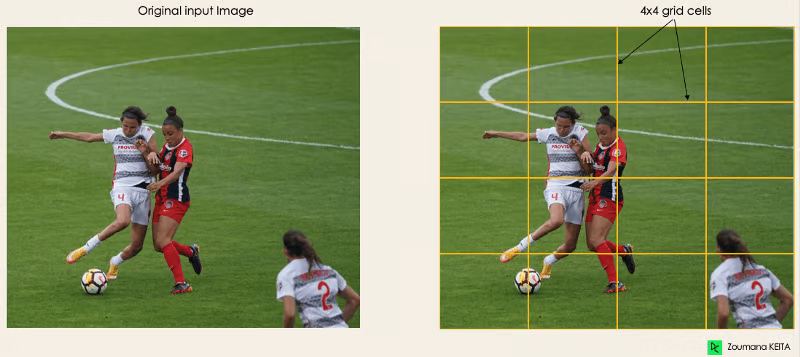
- Đầu tiên, một phép tích chập 1×1 được áp dụng để giảm số lượng kênh, sau đó là một phép tích chập 3×3 để tạo đầu ra hình khối.

- Hàm kích hoạt được sử dụng trong mô hình là ReLU, ngoại trừ lớp cuối cùng, nơi sử dụng hàm kích hoạt tuyến tính.

- Một số kỹ thuật bổ sung như chuẩn hóa batch (batch normalization) và dropout được sử dụng để điều chỉnh mô hình và ngăn chặn hiện tượng overfitting.

### 2.3.2. Quy trình nhận diện của YOLO

*Bước 1: Chia ảnh thành lưới*

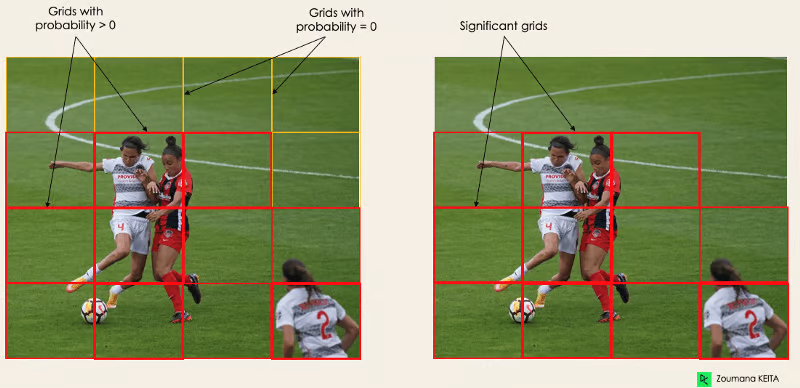
**

Bước đầu tiên bắt đầu bằng cách chia hình ảnh gốc thành các ô lưới NxN có kích thước bằng nhau, trong đó N, trong trường hợp này, là 4, như được hiển thị trong hình trên. Mỗi ô trong lưới có nhiệm vụ xác định vị trí và dự đoán lớp của đối tượng mà nó bao phủ, cùng với giá trị xác suất/tự tin (confidence value).

*Bước 2: Dự đoán Bounding Box và Class Score*

Bước tiếp theo là xác định các hộp giới hạn (bounding boxes) tương ứng với các hình chữ nhật, làm nổi bật tất cả các đối tượng trong hình ảnh. Số lượng hộp giới hạn có thể nhiều như số lượng đối tượng trong một hình ảnh nhất định.

YOLO xác định các thuộc tính của các hộp giới hạn này bằng một mô-đun hồi quy duy nhất theo định dạng sau, trong đó Y là biểu diễn vector cuối cùng cho mỗi hộp giới hạn:



Điều này đặc biệt quan trọng trong giai đoạn huấn luyện mô hình.

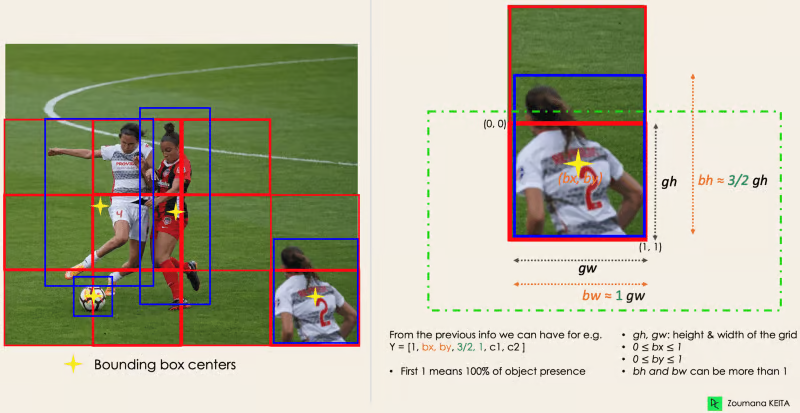
pc là điểm xác suất của ô lưới chứa một đối tượng. Ví dụ, tất cả các ô lưới màu đỏ sẽ có điểm xác suất lớn hơn 0. Hình ảnh trên là phiên bản đơn giản hóa vì xác suất của mỗi ô màu vàng bằng 0 (không đáng kể).

bx, by là tọa độ x và y của tâm hộp giới hạn so với ô lưới bao quanh nó.

bh, bw tương ứng với chiều cao và chiều rộng của hộp giới hạn so với ô lưới bao quanh.

c1 và c2 là hai lớp, ví dụ trong trường hợp này là "Player" và "Ball". Số lượng lớp có thể thay đổi tùy theo yêu cầu của từng bài toán.

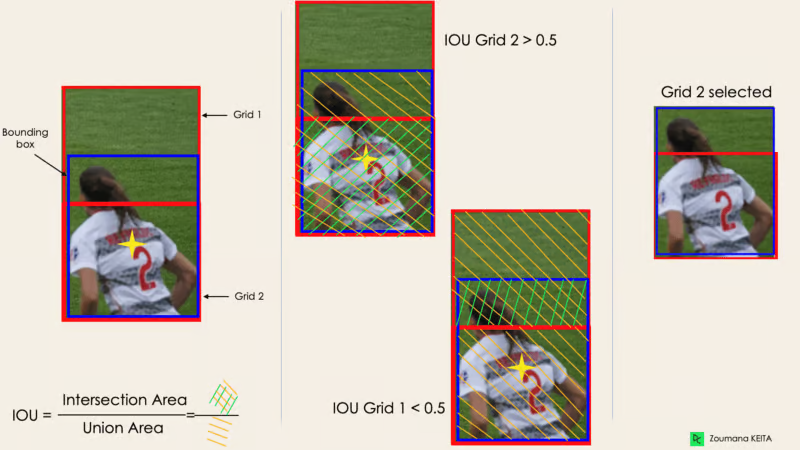
Để hiểu rõ hơn, có thể quan sát hình dưới, chú ý cầu thủ ở góc dưới phải.



*Bước 3: Xử lý hậu kỳ (Post-processing) – Loại bỏ hộp trùng lặp*

Sau khi dự đoán, YOLO có thể tạo ra nhiều hộp giới hạn (bounding boxes) chồng lên nhau cho cùng một vật thể. Do đó, cần một bước xử lý hậu kỳ để lọc ra hộp tốt nhất. Nếu một hộp có *confidence score* < *threshold* (ví dụ: 0.25), thì nó sẽ bị loại bỏ.

Hầu hết các trường hợp, một đối tượng trong hình ảnh có thể có nhiều ô lưới dự đoán khác nhau, mặc dù không phải tất cả đều có ý nghĩa. Mục tiêu của IOU (một giá trị nằm trong khoảng từ 0 đến 1) là loại bỏ các ô lưới không cần thiết và chỉ giữ lại những ô lưới phù hợp.

**

*IoU (Intersection over Union)* đo độ trùng khớp giữa hai box:

Nếu IoU = 1, hai box trùng khít.

Nếu IoU = 0, hai box không chồng lên nhau.

Nếu IoU quá thấp (ví dụ: 0.1) → Có thể giữ lại quá nhiều box → Gây trùng lặp.

Nếu IoU quá cao (ví dụ: 0.7) → Có thể loại bỏ box đúng → Giảm độ chính xác.

*Kết quả*: YOLO giữ lại hộp tốt nhất, giúp mô hình hoạt động nhanh và chính xác hơn.

Việc đặt một ngưỡng cho IOU không phải lúc nào cũng đủ, vì một đối tượng có thể có nhiều hộp giới hạn (bounding boxes) với IOU vượt qua ngưỡng, và giữ lại tất cả những hộp này có thể tạo ra nhiễu.

Đây là lúc chúng ta có thể sử dụng NMS (Non-Maximum Suppression) để chỉ giữ lại các hộp có điểm xác suất phát hiện cao nhất. Non-Maximum Suppression (NMS) là một kỹ thuật được sử dụng trong YOLO để lọc bớt các bounding boxes trùng lặp và chỉ giữ lại hộp có điểm dự đoán cao nhất.

### 2.3.3. Công thức tính toán trong YOLO

#### 2.3.3.1. Công thức tính Confidence Score

*P(Object)*: Xác xuất vật thể có trong ô

*IoU*: Độ trùng khớp giữa box dự đoán và box thật

Nếu box không chứa vật thể nào thì *Confidence Score = 0*

#### 2.3.3.2. Hàm mất mát (Loss Function)

*Localization Loss:* Sai số giữa bounding box dự đoán và bounding box thật.

YOLO sử dụng hàm *MSE (Mean Squared Error)* để đo độ lệch của tọa độ trung tâm (x, y), chiều rộng (w), chiều cao (h).

Vì vật thể lớn cần độ chính xác cao hơn, YOLO lấy căn bậc hai của w, h thay vì giá trị tuyệt đối.

*Confidence Loss:* Sai số giữa độ tin cậy dự đoán và thực tế.

Nếu ô không có vật thể, confidence phải bằng 0.

*Classification Loss*: Sai số giữa nhãn thực tế và nhãn dự đoán của vật thể.

#### 2.3.3.3. Hàm MSE (Mean Square Error)

Hàm MSE (Mean Squared Error) là một trong những hàm mất mát phổ biến nhất trong học sâu và thống kê. Nó được sử dụng để đo lường sự khác biệt giữa giá trị dự đoán của mô hình và giá trị thực tế.

Trong đó:

* : Số lượng mẫu
* : Giá trị thực tế của mẫu thứ *i*
* : Giá trị dự đoán của mẫu thứ *i*
* : Sai số bình phương giữa giá trị thực tế và giá trị dự đoán.

*Ý nghĩa của MSE:*

* MSE càng nhỏ, mô hình càng chính xác
* Vì sai số được bình phương, MSE phạt nặng các lỗi lớn hơn, giúp mô hình tập trung vào việc giảm các sai số lớn.

*Nhược điểm của MSE:*

* Đơn vị của MSE là bình phương đơn vị đầu vào (nếu đầu vào là mét thì MSE có đơn vị mét²), điều này khiến khó diễn giải.
* Dễ bị ảnh hưởng bởi outliers (giá trị ngoại lai).

*Ứng dụng MSE trong YOLO*

* Dùng MSE trên tọa độ *x*, *y* để tối ưu vị trí box.
* Dùng MSE trên căn bậc hai của 𝑤, ℎ để tránh ảnh hưởng quá lớn từ kích thước box lớn.

Nhờ MSE, YOLO có thể học cách dự đoán bounding box chính xác hơn, giảm sai số trong quá trình nhận diện vật thể.

## 2.4. React Native

## 2.5. Thiết bị phần cứng

# CHƯƠNG 3. PHÂN TÍCH – THIẾT KẾ HỆ THỐNG

## 3.1. Phân tích chức năng hệ thống

### 3.1.1. Chức năng của ứng dụng di động ICarP

|  |  |
| --- | --- |
| Người dùng ứng dụng | Quản lý |
| Đăng ký tài khoản | Cập nhật thông tin bãi gửi xe |
| Đăng nhập | Cập nhật thông tin người dùng |
| Xác thực thông tin |  |
| Tìm kiếm bãi gửi xe |  |
| Nạp tiền |  |
| Liên kết tài khoản ngân hàng |  |
| Thanh toán hóa đơn |  |
| Xuất mã QR |  |
| Chỉnh sửa thông tin phương tiện |  |
| Chỉnh sửa thông tin cá nhân |  |
| Xem lịch sử gửi xe |  |

3.1.2. Chức năng của hệ thống bãi gửi xe thông minh ICarP

|  |  |
| --- | --- |
| Người dùng | Quản lý |
| Quét mã check in | Lập báo cáo |
| Quét mã check out |  |
| Xuất hóa đơn |  |
| Quét mã thanh toán |  |
|  |  |

3.2. Thiết kế hệ thống

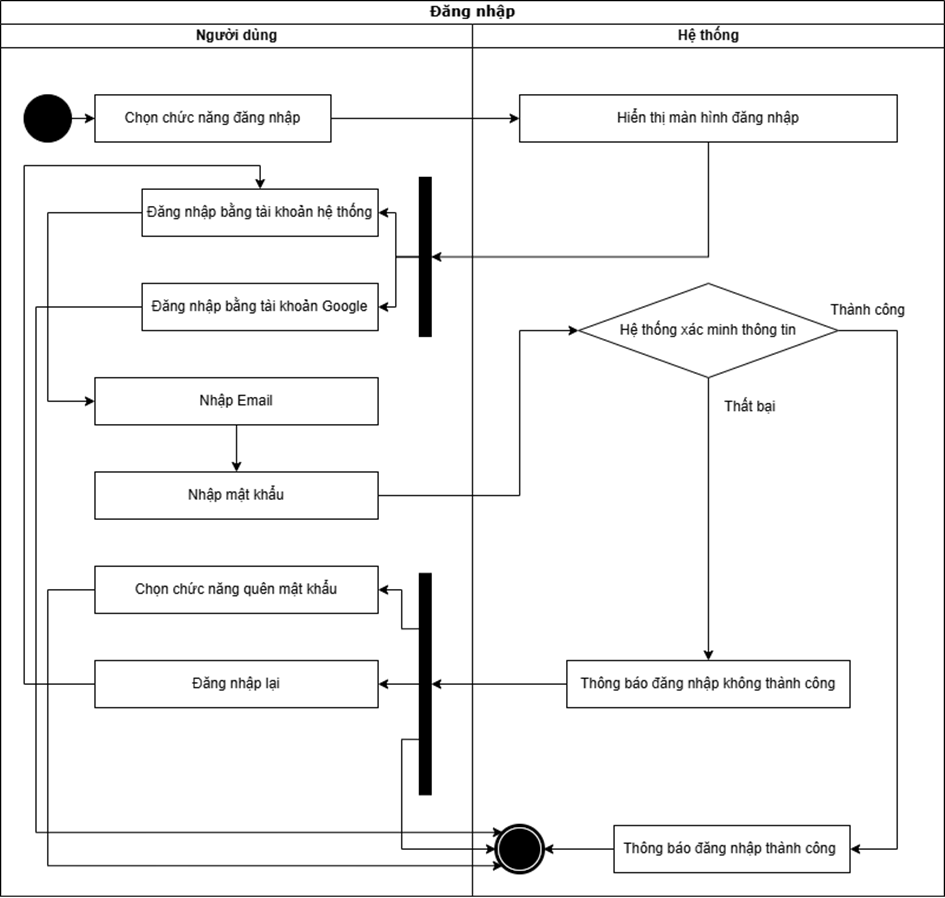
### 3.2.1. Ứng dụng di động ICarP

*Biểu đồ use-case tổng quát*

*A diagram of a diagram

AI-generated content may be incorrect.*

*Biểu đồ hoạt động chức năng đăng nhập*

**

*Biểu đồ hoạt động chức năng đăng ký*

A diagram with black lines and white text

AI-generated content may be incorrect.

*Biểu đồ chức năng xét duyệt hồ sơ thông tin cá nhân*

A diagram of a work flow

AI-generated content may be incorrect.

*Biểu đồ chức năng xét duyệt hồ sơ phương tiện*

*A diagram of a diagram

AI-generated content may be incorrect.*

*Biểu đồ hoạt động chức năng tìm kiếm bãi gửi xe*

A diagram of a diagram

AI-generated content may be incorrect.

*Biểu đồ hoạt động chức năng xem lịch sử gửi xe*

A diagram with text and images

AI-generated content may be incorrect.

*Biểu đồ hoạt động chức năng nạp tiền*

A diagram of a company

AI-generated content may be incorrect.

*Biểu đồ hoạt động chức năng liên hệ, báo cáo sự cố*

*A screenshot of a computer

AI-generated content may be incorrect.*

*Biểu đồ hoạt động chức năng quét mã check in*

*A screenshot of a diagram

AI-generated content may be incorrect.*

*Biểu đồ hoạt động chức năng quét mã check out*

*A screenshot of a diagram

AI-generated content may be incorrect.*

*Biểu đồ hoạt động chức năng xét duyệt hồ sơ cá nhân và hồ sơ phương tiện*

*A screenshot of a diagram

AI-generated content may be incorrect.*

### 3.2.2. Hệ thống bãi gửi xe thông minh ICarP

*Biểu đồ Use-case tổng quát*

*Biểu đồ chức năng trích xuất biển số phương tiện*

A diagram with black text

AI-generated content may be incorrect.

CHƯƠNG………

Nội dung đề mục

Nội dung nhóm tiểu mục

Nội dung nhóm tiểu mục

Nội dung tiểu mục

Nội dung tiểu mục

# KẾT LUẬN

# TÀI LIỆU THAM KHẢO

*(Cách trình bày tài liệu tham khảo được trích dẫn theo IEEE)*

1. Bộ Nông nghiệp và Phát triển Nông thôn, Thông tư số 18/2014/TT-BNNPTNT ngày 23/6/2014 ban hành Danh mục bổ sung giống vật nuôi được phép sản xuất, kinh doanh tại Việt Nam, 2014.
2. L. M. Gaetke and C. K.Chow, “Copper toxicity, oxidative stress, and antioxidant nutrients,” Toxicology, Vol. 189, No. 1–2, pp.147–163, 2003. DOI: 10.1016/S0300- 483X(03)00159-8.
3. N. T. L Hương và T. T. Quân, “Nhận thức của du khách về hình ảnh điểm đến du lịch Huế,” Tạp chí Khoa học Đại học Huế: Kinh tế và Phát triển, Tập 126, Số 5D, tr. 79– 94, 2017. DOI: 10.26459/hueuni-jed.v126i5D.4555.
4. L.T.K. Liên, T. T. T. Thủy, Q. B. Chính và T. N. Quyền, “Đánh giá của du khách về du lịch lễ hội tổ chức tại chùa ở Thừa Thiên Huế,” Tạp chí Khoa học Đại học Huế, Tập 109, Số 10, tr. 191–202, 2015.
5. P. K. Liệu và T. A. Tuấn, “Tính toán mức phát thải nhà kính của chính quyền thành phố Huế bằng công cụ Bilan Carbone,” trong Kỷ yếu Hôi thảo Khoa học Quốc gia Đất ngập nước và Biến đổi khí hậu, Hà Nội, 2011, Hà Nội: Nxb Khoa học và Kỹ thuật, 2011, tr. 343-356.
6. L. V. Mỹ, “Ngoại giao Cộng hòa Nhân dân Trung Hoa 30 năm cải cách mở cửa (1978-2008)”. Hà Nội: Nxb Khoa học Xã hội, 2007.
7. T. Tamminen, “Eutrophication and the Baltic Sea: Studies on Phytoplankton, Bacterioplankton and Pelagic Nutrient Cycles,” PhD thesis, University of Helsinki, Finland, 1990.
8. N. Q. T. Tiến, “Về quá trình tụ cư lập làng ở Hương Vinh,” trong Văn hóa - lịch sử Huế qua góc nhìn làng xã phụ cận và quan hệ với bên ngoài, N. Q. T. Tiến và N. Masanari, Chủ biên. Huế: Nxb.Thuận Hóa, 2010, tr.10 - 28.

# PHỤ LỤC

**Phụ lục A: Tên nội dung phụ lục**

***Phụ lục A.1: Tên nội dung phụ lục***

**Hình A.1:**

**Bảng A.1:**

***Phụ lục A.2: Tên nội dung phụ lục***

**Phụ lục B: Tên nội dung phụ lục**

***Phụ lục B.1: Tên nội dung phụ lục***

**Hình B.1:**

**Bảng B.1:**

***Phụ lục B.2: Tên nội dung phụ lục***

# Kết quả turnitin

*(In trang đầu tiên trong báo cáo check turnitin và đóng vào cuối quyển)*

**QUI CÁCH ĐÓNG QUYỂN KHÓA LUẬN TỐT NGHIỆP**

