**ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**---------------------------------------**

**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

NGHIÊN CỨU VÀ ỨNG DỤNG MÔ HÌNH MOBILENETV2 TRÊN NỀN TẢNG DI ĐỘNG TRONG BÀI TOÁN PHÂN LOẠI HOA QUẢ

**GVHD: TS. Nguyễn Văn Tỉnh**

**Sinh viên: Nguyễn Hưng Thịnh**

**Mã sinh viên: 2021603670**

**Lớp: KHMT1 Khóa: K16**

Hà Nội – 2025

# MỤC LỤC

[MỤC LỤC 2](#_Toc196294708)

[DANH MỤC HÌNH ẢNH 4](#_Toc196294709)

[DANH MỤC BẢNG BIỂU 5](#_Toc196294710)

[LỜI CẢM ƠN 6](#_Toc196294711)

[LỜI NÓI ĐẦU 1](#_Toc196294712)

[CHƯƠNG 1. TỔNG QUAN VỀ LÝ THUYẾT 3](#_Toc196294713)

[1.1. Tổng quan về học máy 3](#_Toc196294714)

[1.1.1. Phân loại học máy 4](#_Toc196294715)

[1.1.2. Các bước huấn luyện mô hình học máy 7](#_Toc196294716)

[1.1.3. Ứng dụng của học máy 8](#_Toc196294717)

[1.2. Tổng quan về CNN 9](#_Toc196294718)

[1.3. Tổng quan về Dart và framework Flutter 12](#_Toc196294719)

[1.3.1. Tổng quan về Dart 13](#_Toc196294720)

[1.3.2. Tổng quan về flutter 14](#_Toc196294721)

[1.4. Giới thiệu về công cụ 15](#_Toc196294722)

[1.4.1. Android Studio 15](#_Toc196294723)

[1.4.2. Pycharm 16](#_Toc196294724)

[1.4.3. TensorFlow 18](#_Toc196294725)

[CHƯƠNG 2. THIẾT KẾ HỆ THỐNG VÀ MÔI TRƯỜNG 19](#_Toc196294726)

[2.1. Cấu trúc mô hình MobileNetV2 19](#_Toc196294727)

[2.2. Chuẩn bị và phân tích dữ liệu 20](#_Toc196294728)

[2.3. Tiền xử lý dữ liệu 21](#_Toc196294729)

[CHƯƠNG 3. CÀI ĐẶT HỆ THỐNG VÀ MỘT SỐ KẾT QUẢ 23](#_Toc196294730)

[3.1. Huấn luyện mô hình 23](#_Toc196294731)

[3.2. Phân tích mô hình 23](#_Toc196294732)

[CHƯƠNG 4: TRIỂN KHAI ỨNG DỤNG DI ĐỘNG 25](#_Toc196294733)

[4.1. Các màn hình trong ứng dụng 25](#_Toc196294734)

[4.2. Thử nghiệm trên thiết bị thực tế 25](#_Toc196294735)

[KẾT LUẬN 26](#_Toc196294736)

[TÀI LIỆU THAM KHẢO 27](#_Toc196294737)

# DANH MỤC HÌNH ẢNH

[Hình 1.1. Mối quan hệ giữa học máy, học sâu và trí tuệ nhân tạo 3](#_Toc196294738)

[Hình 1.2. Sơ đồ hoạt động học có giám sát. Ảnh: Supervised Learning và Unsupervised Learning: Khác biệt là gì? - VinBigdata - Blog 5](#_Toc196294739)

[Hình 1.3. Sơ đồ hoant động học không giám sát. Ảnh: Supervised Learning và Unsupervised Learning: Khác biệt là gì? - VinBigdata - Blog 6](#_Toc196294740)

[Hình 1.4. Thuật toán học tăng cường. Ảnh: Học tăng cường là gì? Lợi ích học tăng cường manh lại 7](#_Toc196294741)

[Hình 1.5. Kiến trúc các lớp CNN 9](#_Toc196294742)

[Hình 1.6. Kiến trúc mạng VGG16 11](#_Toc196294743)

[Hình 1.7. Kiến trúc mạng InceptionV3 12](#_Toc196294744)

[Hình 1.8. Kiến trúc mạng ResNet50 12](#_Toc196294745)

[Hình 1.9. Biểu tượng của ngôn ngữ Dart 13](#_Toc196294746)

[Hình 1.10. Biểu tượng của flutter 14](#_Toc196294747)

[Hình 1.11. Biểu tượng android studio 15](#_Toc196294748)

[Hình 1.12. Biểu tượng pycharm 17](#_Toc196294749)

# DANH MỤC BẢNG BIỂU

# LỜI CẢM ƠN

Với tình cảm sâu sắc và chân thành nhất, em xin được phép bày tỏ lòng biết ơn đến tất cả các cá nhân và tổ chức đã tạo điều kiện hỗ trợ, giúp đỡ em trong suốt quá trình học tập và nghiên cứu đề tài “*Nghiên cứu và ứng dụng mô hình MobileNetV2 trong bài toán phân loại hoa quả*”. Trong suốt thời gian từ khi bắt đầu học tập tại trường đến nay, em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý thầy cô và bạn bè. Với lòng biết ơn sâu sắc nhất, em xin gửi đến quý thầy cô ở khoa công nghệ thông tin đã truyền đạt vốn kiến thức quý báu cho em trong suốt thời gian học tập tại trường. Nhờ có những lời hướng dẫn, dạy bảo của các thầy, các cô nên đề tài tốt nghiệp: “*Nghiên cứu và ứng dụng mô hình MobileNetV2 trong bài toán phân loại hoa quả*” của em mới có thể hoàn thiện. Một lần nữa, em xin chân thành cảm ơn thầy Nguyễn Văn Tỉnh – người đã trực tiếp giúp đỡ, quan tâm, hướng dẫn em hoàn thành tốt bài báo cáo này trong thời gian qua. Bài báo cáo đồ án của em thực hiện trong khoảng thời gian 9 tuần. Bước đầu đi vào thực tế của em còn hạn chế và còn nhiều bỡ ngỡ nên không tránh khỏi những thiếu sót, em rất mong nhận được những ý kiến đóng góp quý báu của quý thầy cô để kiến thức của em trong lĩnh vực này được hoàn thiện hơn đồng thời có điều kiện bổ sung, nâng cao kỹ năng nghề nghiệp của em.

Em xin chân thành cảm ơn!

Nguyễn Hưng Thịnh

# LỜI NÓI ĐẦU

**Chương 1:**

**Chương 2:**

**Chương 3:**

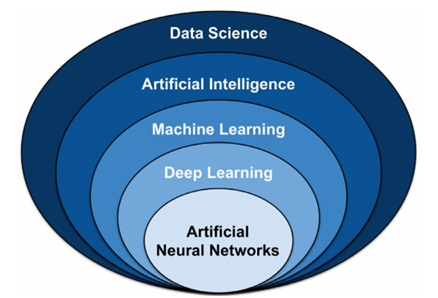
# CHƯƠNG 1. TỔNG QUAN VỀ LÝ THUYẾT

## Tổng quan về học máy

Học máy (Machine learning) là một lĩnh vực tập trung vào khía cạnh "học" của trí tuệ nhân tạo (Artificial intelligence) bằng cách phát triển các thuật toán có thể mô tả tốt nhất một tập dữ liệu.

Giả sử một công ty bất động sản muốn dự đoán giá của một căn nhà dựa trên các đặc điểm cụ thể của căn nhà đó. Trước tiên, công ty sẽ thu thập một tập dữ liệu bao gồm nhiều mẫu dữ liệu. Mỗi mẫu đại diện cho một quan sát duy nhất về một căn nhà cùng với các đặc điểm liên quan. Những đặc điểm này là các thuộc tính đã được ghi nhận của căn nhà, và có thể hữu ích để dự đoán giá (ví dụ: diện tích tổng thể, số tầng, có sân vườn hay không). **Target (mục tiêu)** là thuộc tính mà mô hình cần dự đoán — trong trường hợp này là **giá nhà**.

Thông thường, tập dữ liệu sẽ được chia thành ba phần: **training set (tập huấn luyện)**, **validation set (tập xác thực)** và **test set (tập kiểm tra)**, vì mô hình thường sẽ hoạt động tốt nhất trên dữ liệu mà nó đã được huấn luyện.

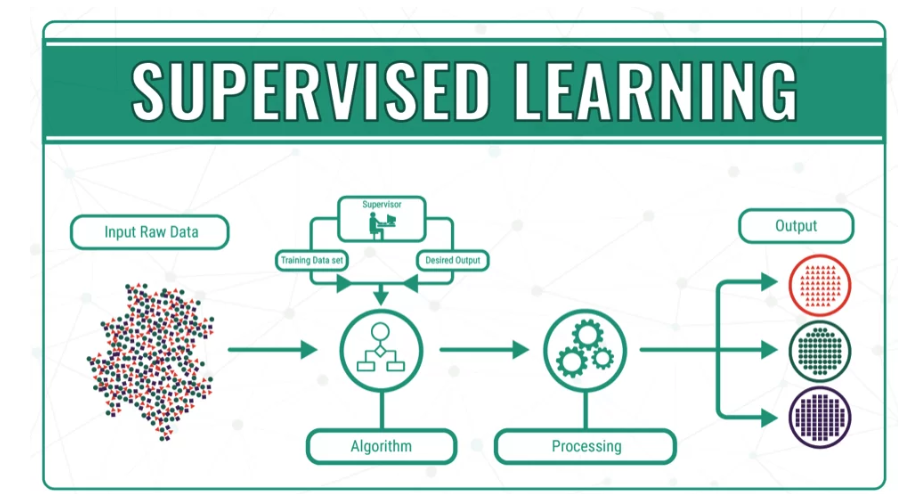


Hình 1.1. Mối quan hệ giữa học máy, học sâu và trí tuệ nhân tạo

### Phân loại học máy

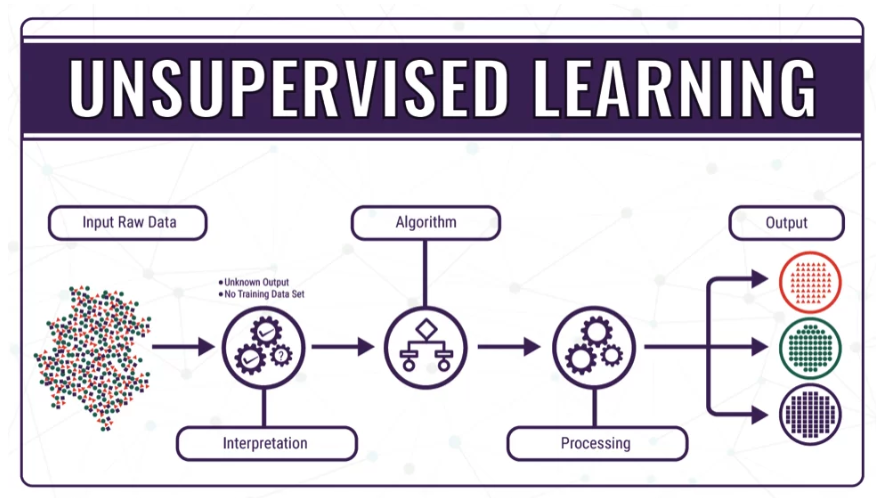
Học máy có thể được phân loại thành ba nhóm chính dựa trên cách thức dữ liệu được sử dụng trong quá trình huấn luyện và mục tiêu mà mô hình hướng đến. Ba nhóm chính này bao gồm **học giám sát** (supervised learning), **học không giám sát** (unsupervised learning), và **học tăng cường** (reinforcement learning). Mỗi loại có cách tiếp cận và ứng dụng khác nhau, tạo ra những điểm mạnh và hạn chế riêng biệt, tùy thuộc vào bài toán và dữ liệu cụ thể.

* Học có giám sát: Học giám sát là một trong những phương pháp phổ biến và cơ bản trong học máy. Trong phương pháp này, mô hình học từ dữ liệu có nhãn (labeled data), tức là mỗi ví dụ trong bộ dữ liệu huấn luyện có một nhãn đầu ra đã biết. Mục tiêu là huấn luyện một mô hình sao cho khi đưa vào dữ liệu mới, mô hình có thể dự đoán đúng nhãn đầu ra cho dữ liệu chưa được gắn nhãn
* **Quá trình huấn luyện học có giám sát**: Quá trình huấn luyện trong học giám sát thường bao gồm việc tối ưu hóa một hàm mất mát (loss function), trong đó mô hình cố gắng dự đoán nhãn đầu ra sao cho độ chệch giữa dự đoán và nhãn thực tế là nhỏ nhất.
* **Ứng dụng:** Học giám sát rất hữu ích trong các bài toán phân loại và hồi quy. Các bài toán phân loại yêu cầu phân nhóm dữ liệu vào các lớp khác nhau (ví dụ: nhận diện ảnh của các loài động vật). Các bài toán hồi quy yêu cầu dự đoán giá trị liên tục (ví dụ: dự đoán giá trị bất động sản, nhiệt độ, hay chỉ số chứng khoán).
* **Các thuật toán phổ biển:** Hồi quy tuyến tính, SVM (Supporrt vector machine), K lân cận (KNN),…



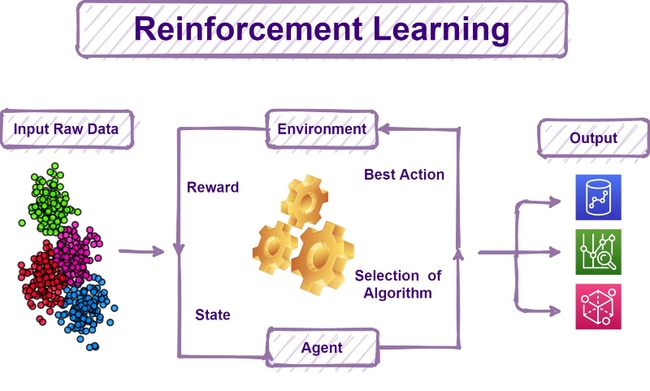
Hình 1.2. Sơ đồ hoạt động học có giám sát. Ảnh: [Supervised Learning và Unsupervised Learning: Khác biệt là gì? - VinBigdata - Blog](https://blog.vinbigdata.org/supervised-learning-va-unsupervised-learning-khac-biet-la-gi/)

* Học không giám sát: Khác với học giám sát, học không giám sát sử dụng dữ liệu không có nhãn. Mục tiêu của học không giám sát là tìm ra các cấu trúc, mẫu hoặc sự phân bố ẩn trong dữ liệu mà không cần sự giám sát từ người dùng. Điều này đặc biệt hữu ích khi không có sẵn dữ liệu đã được gắn nhãn hoặc khi chúng ta muốn khám phá các đặc trưng tiềm ẩn trong tập dữ liệu.
* **Quá trình huấn luyện học không giám sát**: Trong học không giám sát, mô hình học tự động nhận diện các mẫu trong dữ liệu mà không cần biết trước các nhóm hay nhãn. Mô hình sẽ học cách phân nhóm hoặc giảm chiều dữ liệu để tìm ra các mối quan hệ giữa các dữ liệu.
* **Ứng dụng**: Học không giám sát thường được sử dụng trong các bài toán phân cụm, giảm chiều dữ liệu, và phát hiện bất thường. Ví dụ, trong phân tích dữ liệu khách hàng, học không giám sát có thể giúp phát hiện các nhóm khách hàng có hành vi tương tự. Trong xử lý ảnh, các kỹ thuật này có thể được sử dụng để giảm số lượng đặc trưng mà vẫn giữ lại các thông tin quan trọng.
* **Các thuật toán phổ biến**: Phân cụm K-means (K-means clustering), Naïve Bayes,…



Hình 1.3. Sơ đồ hoạt động học không giám sát. Ảnh: [Supervised Learning và Unsupervised Learning: Khác biệt là gì? - VinBigdata - Blog](https://blog.vinbigdata.org/supervised-learning-va-unsupervised-learning-khac-biet-la-gi/)

* Học tăng cường: Học tăng cường là một phương pháp học máy mà trong đó một tác nhân (agent) học cách đưa ra các quyết định thông qua tương tác với môi trường và nhận phản hồi dưới dạng thưởng (reward) hoặc phạt (punishment). Mục tiêu của tác nhân là tối đa hóa tổng hợp phần thưởng trong suốt quá trình tương tác, qua đó học được cách hành động tối ưu.
* **Quá trình huấn luyện**: Trong học tăng cường, tác nhân sẽ thực hiện các hành động và nhận phần thưởng từ môi trường. Thông qua thử nghiệm và sai sót, tác nhân dần dần tìm ra chiến lược tốt nhất để đạt được mục tiêu dài hạn. Phương pháp này yêu cầu sự kết hợp giữa việc khai thác (exploitation) các kiến thức hiện tại và khám phá (exploration) các chiến lược mới.
* **Ứng dụng**: Học tăng cường có ứng dụng nổi bật trong các trò chơi, robot học, xe tự lái, và các hệ thống tối ưu hóa. Một trong những ví dụ thành công nhất của học tăng cường là AlphaGo của Google DeepMind, nơi AI đã học cách chơi cờ vây và đánh bại các kỳ thủ giỏi nhất thế giới.
* **Các thuật toán phổ biến**: Deep Q-Networks (DQN), Policy Gradient, …



Hình 1.4. Thuật toán học tăng cường. Ảnh: [Học tăng cường là gì? Lợi ích học tăng cường manh lại](https://bizflycloud.vn/tin-tuc/hoc-tang-cuong-la-gi-20240410141541697.htm)

### Các bước huấn luyện mô hình học máy

**Bước 1:** Thu thập và xử lý dữ liệu

Dữ liệu là yếu tố quan trọng trong học máy. Thu thập và xử lý dữ liệu là bước đầu tiên và quan trọng nhất. Dữ liệu phải được thu thập từ các nguồn đáng tin cậy và được tiền xử lý để đảm bảo rằng chúng có chất lượng cao và sẵn sàng để huấn luyện mô hình. Tiền xử lý dữ liệu bao gồm:

* Làm sạch dữ liệu
* Chuyển đổi dữ liệu về định dạng phù hợp
* Chuẩn hoá dữ liệu để cải thiện hiệu suất mô hình

**Bước 2:** Lựa chọn mô hình phù hợp cho bài toán

Lựa chọn mô hình học máy phù hợp với bài toán là rất quan trọng. Các mô hình khác nhau sẽ có hiệu suất khác nhau tùy thuộc vào bản chất của dữ liệu và bài toán cần giải quyết. Ví dụ, nếu bài toán yêu cầu phân loại ảnh, các mô hình học sâu như mạng nơ-ron tích chập (Convolutional Neural Networks - CNNs) sẽ hiệu quả hơn.

**Bước 3:** Huấn luyện mô hình

Sau khi chuẩn bị dữ liệu và chọn mô hình, bước tiếp theo là huấn luyện mô hình. Quá trình huấn luyện liên quan đến việc tối ưu hóa các tham số của mô hình sao cho mô hình có thể dự đoán chính xác nhất trên dữ liệu huấn luyện.

**Bước 4:** Đánh giá mô hình

Sau khi huấn luyện, mô hình cần được đánh giá để kiểm tra độ chính xác và khả năng tổng quát của nó trên dữ liệu chưa thấy trước đó. Các chỉ số đánh giá phổ biến bao gồm:

* Độ chính xác (Accuracy)
* Độ chính xác trung bình
* Mất mát (Loss)
* Độ nhạy và độ đặc hiệu

### Ứng dụng của học máy

Học máy đang ngày càng trở thành một công cụ thiết yếu trong thời đại số, với sự ứng dụng rộng rãi trong hầu hết mọi lĩnh vực của đời sống hiện đại. Các thuật toán học máy giúp tự động hóa quá trình ra quyết định, phát hiện mẫu, và xử lý lượng dữ liệu khổng lồ mà con người không thể làm được một cách thủ công. Dưới đây là một số lĩnh vực tiêu biểu mà học máy đã và đang được ứng dụng mạnh mẽ:

* Thị giác máy tính: Nhận diện ảnh, phân loại ảnh, nhận dạng ký tự và biển số xe, chẩn đoán y tế từ ảnh, …
* Xử lý ngôn ngữ tự nhiên: Dịch máy, chatbot và trợ lý ảo, phân tích cảm xúc, tóm tắt văn bản tự động, …
* Tài chính ngân hàng: Phát hiện gian lận, dự báo tài chính, đánh giá tín dụng, tư vấn tài chính cá nhân,…
* Giao thông và phương tiện tự hành: Xe tự đoán, dự đoán lưu lượng giao thông, hệ thống giao thông thông minh, ..

## Tổng quan về CNN

Trong lĩnh vực thị giác máy tính, mạng nơ-ron tích chập (CNN) về cơ bản là một mạng perceptron đa lớp (MLP) được chuẩn hóa và đã trở thành một trong những đổi mới có ảnh hưởng nhất. Trong những năm gần đây, CNN đã thống trị lĩnh vực thị giác máy tính và xử lý hình ảnh cho các tác vụ nhận dạng, phân loại và phân đoạn hình ảnh quy mô lớn. Một CNN điển hình bắt đầu với một lớp đầu vào và kết thúc với một lớp đầu ra, giữa chúng là nhiều lớp ẩn. Các lớp tích chập, pooling, chuẩn hóa (ReLU), và lớp kết nối đầy đủ là một phần của lớp ẩn. Hình 1.1 dưới đây cho thấy các lớp CNN điển hình.

Diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a

Description automatically generated

Hình 1.5. Kiến trúc các lớp CNN

Lớp đầu vào nhận dữ liệu hình ảnh mục tiêu làm đầu vào. Hình ảnh sau đó được thay đổi kích thước tối ưu và chuyển tiếp đến lớp tiếp theo là lớp tích chập. Tại đây, tồn tại một số lượng hạt nhân hoặc bộ lọc thực hiện phép nhân theo từng phần tử để trích xuất đặc trưng. Thông qua hàm kích hoạt, giá trị âm của đầu vào sẽ được thay thế bằng không, nếu không nó sẽ được chuyển trực tiếp đến đầu ra. Hàm kích hoạt được sử dụng rộng rãi nhất là ReLU (Rectified Linear Unit) và đối với nhiều loại mạng nơ-ron, nó là hàm kích hoạt mặc định. Đây là một hàm phi tuyến và nhanh hơn các hàm kích hoạt khác như Sigmoid, SELU (Scaled Exponential Linear Unit), ELU (Exponential Linear Unit), GELU (Gaussian Error Linear Unit) v.v. Các đặc trưng được trích xuất từ lớp tích chập sau đó được gửi đến lớp pooling. Lớp này chỉ giữ lại các đặc trưng quan trọng từ một hình ảnh lớn bằng cách giảm số lượng tham số. Sau đó, lớp kết nối đầy đủ dịch các hình ảnh đã được lọc kỹ lưỡng này thành các danh mục. Và một hàm phi tuyến khác tên là softmax cuối cùng sẽ cung cấp các xác suất thập phân từ 0 đến 1 cho mỗi lớp. Trong nghiên cứu này, một mạng nơ-ron tích chập 6 lớp được đề xuất và được xây dựng hoàn toàn từ đầu. Kích thước hình ảnh đầu vào được chọn là 32×32 để giảm thời gian tính toán tổng thể, giúp tạo ra một mô hình tốt về mặt hiệu quả. Các kỹ thuật tăng cường dữ liệu như xoay, thay đổi tỷ lệ, cắt xén, phóng to và lật ngang cũng được áp dụng cho dữ liệu hình ảnh huấn luyện có kích thước 32×32 và 3 kênh. ReLU đã được sử dụng làm hàm kích hoạt cho mỗi lớp tích chập. Và để cải thiện lỗi tổng quát hóa, một tỷ lệ dropout là 0,25 được sử dụng để khắc phục các vấn đề overfitting. Cuối cùng, softmax được sử dụng để tìm các xác suất của mỗi lớp dưới dạng số thập phân. Hình 1.2 cho thấy kiến trúc mô hình CNN đã được phát triển.

Các mô hình CNN phổ biến

* **Mô hình VGG16:** VGG16 là một kiến trúc mạng nơ-ron liên kết (CNN) đơn giản và được sử dụng rộng rãi được sử dụng cho ImageNet, một dự án cơ sở dữ liệu trực quan lớn được sử dụng trong nghiên cứu phần mềm nhận dạng đối tượng trực quan. Kiến trúc VGG16 được phát triển và giới thiệu bởi Karen Simonyan và Andrew Zisserman từ Đại học Oxford, vào năm 2014, thông qua bài báo của họ “Các mạng kết hợp rất sâu để nhận dạng hình ảnh quy mô lớn”. 'VGG' là tên viết tắt của Visual Geometry Group, là một nhóm các nhà nghiên cứu tại Đại học Oxford, những người đã phát triển kiến ​​trúc này, và '16' ngụ ý rằng kiến ​​trúc này có 16 lớp

A diagram of a graph

Description automatically generated

Hình 1.6. Kiến trúc mạng VGG16

* **InceptionV3:** InceptionV3 là một phiên bản cải tiến của kiến trúc Inception, được phát triển bởi nhóm nghiên cứu của Google. Mô hình này được giới thiệu lần đầu tiên trong bài báo khoa học có tên "Rethinking the Inception Architecture for Computer Vision," do Christian Szegedy và các cộng sự công bố vào năm 2015. InceptionV3 là một trong những bước phát triển tiếp theo của dòng Inception, được xây dựng dựa trên các phiên bản trước đó như InceptionV1 (GoogLeNet) và InceptionV2. Mô hình này đã tích hợp nhiều kỹ thuật tiên tiến để cải thiện hiệu suất và giảm số lượng tham số, như việc sử dụng các lớp tích chập phân nhánh (factorized convolutions), kỹ thuật batch normalization, và việc áp dụng các module Inception nhiều lớp. Mô hình InceptionV3 đã đạt được thành công đáng kể trong cuộc thi ILSVRC (ImageNet Large Scale Visual Recognition Challenge), nơi nó được sử dụng để phân loại hình ảnh với độ chính xác cao. Nó đã trở thành một trong những kiến trúc phổ biến nhất cho các nhiệm vụ liên quan đến nhận dạng hình ảnh và được ứng dụng rộng rãi trong nhiều lĩnh vực, từ thị giác máy tính cho đến học sâu.

A diagram of a block diagram

Description automatically generated

Hình 1.7. Kiến trúc mạng InceptionV3

* **ResNet50:** ResNet là một cấu trúc mạng do He Kaiming, Sun Jian và những người khác của Microsoft Research Asia đề xuất vào năm 2015 và đã giành được vị trí đầu tiên trong nhiệm vụ phân loại ILSVRC-2015. Đồng thời, nó đã giành được vị trí đầu tiên trong các nhiệm vụ phát hiện ImageNet, bản địa hóa ImageNet, phát hiện COCO và phân đoạn COCO.

A diagram of a block diagram

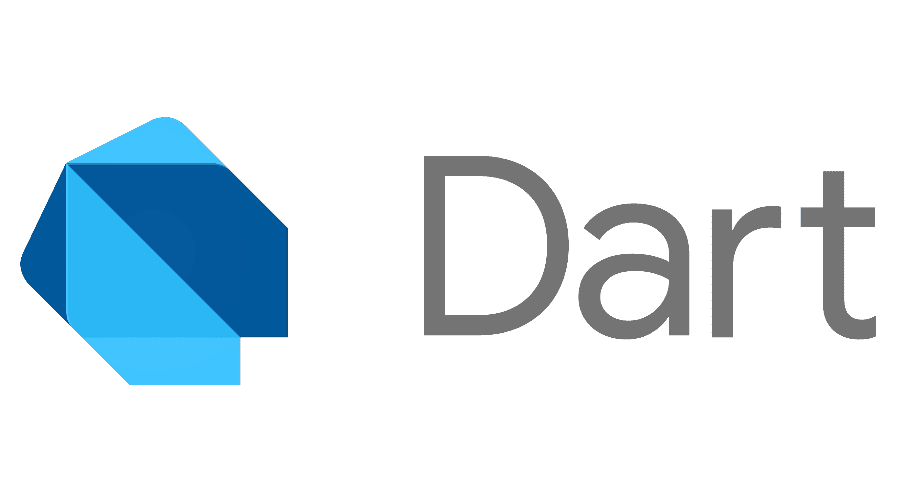
Description automatically generated

Hình 1.8. Kiến trúc mạng ResNet50

* **MobileNetV2:** MobileNetV2 là một kiến trúc mạng nơ-ron tích chập (CNN) được phát triển bởi nhóm nghiên cứu của Google, cụ thể là nhóm Google Research – Brain Team. Kiến trúc này được công bố lần đầu tiên trong bài báo: "MobileNetV2: Inverted Residuals and Linear Bottlenecks". Tác giả: Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen. Được công bố năm 2018, tại hội nghị CVPR (Conference on Computer Vision and Pattern Recognition).

## Tổng quan về Dart và framework Flutter

### Tổng quan về Dart



Hình 1.9. Biểu tượng của ngôn ngữ Dart

Dart, là một ngôn ngữ lập trình đa mục đích mã nguồn mở, đã được đặt nền móng bởi Google. Đây là một khía cạnh của ngôn ngữ lập trình hướng đối tượng, đặc trưng bởi cú pháp kiểu C. Tinh thần lập trình hướng đối tượng thể hiện qua việc hỗ trợ giao diện và lớp, mở ra khả năng sáng tạo không giới hạn khi đặt lên bàn cân với những ngôn ngữ khác. Sự đa dạng của Dart thể hiện thông qua khả năng phát triển ứng dụng web, di động, máy chủ và máy tính để bàn.

Trong lĩnh vực lập trình, Dart là đứa con tâm huyết từ Lars Bak và Kasper Lund, ra đời dưới sự quản lý của Google. Bắt đầu từ lần ra mắt tại hội nghị GOTO tại Đan Mạch vào ngày 10 tháng 12 tháng 10 năm 2011, Dart đã chứng tỏ tiềm năng vô hạn của mình. Một cột mốc quan trọng được ghi dấu bằng việc phát hành phiên bản Dart 1.0 vào ngày 14 tháng 11 năm 2013. Mặc dù ban đầu, Dart nhận được những ý kiến trái chiều, nhưng những phản hồi này không thể ngăn cản sự tiến bộ vượt trội của Dart.

Quãng thời gian này còn đánh dấu sự chuyển mình từ kế hoạch máy ảo Dart trong Chrome, sang việc biên dịch mã Dart thành JavaScript, điều quan trọng mà phiên bản Dart 2.0 đã mang lại vào tháng 8 năm 2018, cùng với những thay đổi ngôn ngữ kể cả hệ thống kiểu.

### Tổng quan về flutter



Hình 1.10. Biểu tượng của flutter

Flutter đã thu hút được sự chú ý của cộng đồng các nhà phát triển bằng cách giới thiệu các style cho phép việc xây dựng UI đẹp hơn và biểu cảm hơn vì thế mà việc code cũng trở nên thú vị hơn nhiều. Nó kết hợp một số khái niệm quen thuộc với những kinh nghiệm phát triển hiện đại như lập trình reactive và widget composition trong khi sử dụng nền tảng Dart làm cơ sở chính cho các hoạt động đó. Nhóm Flutter đã đánh giá nhiều ngôn ngữ khác nhau và cuối cùng họ chọn Dart vì nó phù hợp với cách mà họ xây dựng giao diện người dùng

Flutter có tính năng hot reload giúp bạn thử nghiệm, xây dựng UI, thêm tính năng và fix bug một cách nhanh chóng và dễ dàng hơn. Tính năng hot reload hoạt động bằng cách đưa những file source code đã được update vào máy ảo Virtual Machine (VM) của Dart. Sau khi VM cập nhập các class với phiên bản mới nhất của field và function, framework Flutter sẽ tự động tái xây dựng cây widget, cho phép bạn xem các hiệu ứng mà bạn đã thay đổi một cách nhanh chóng hơn.

Flutter cung cấp phân tích static cho phép bạn phát hiện ra những vấn đề ngay trước khi dòng code đó được thực thi. Nó thực sự là một công cụ mạnh mẽ có thể giúp các dev tránh việc phát sinh bug và đảm bảo được code tuân theo các quy tắc của style.

Ngôn ngữ Dart có trình biên dịch AOT (Ahead of Time) giúp nó biên dịch nhanh, đúng và native code. Điều này không chỉ đảm bảo giúp Flutter nhanh hơn mà còn đảm bảo rằng hầu như mọi thứ (bao gồm tất cả các widget) đều có thể được tùy chỉnh. Với lý do đó cho nên hầu hết các phần của Flutter đều được viết bằng ngôn ngữ này.

Với tất cả những cột mốc thành tựu, không thể phủ nhận vai trò và tiềm năng của Dart trong giới lập trình. Dù đã có những bước tiến, Dart vẫn tiếp tục hướng tới tương lai với sự đổi mới và sáng tạo không ngừng. Như một cây cầu nối giữa ý tưởng và hiện thực, Dart tiếp tục mở ra cơ hội cho sự sáng tạo và phát triển trong tương lai của lập trình.

## Giới thiệu về công cụ

### Android Studio



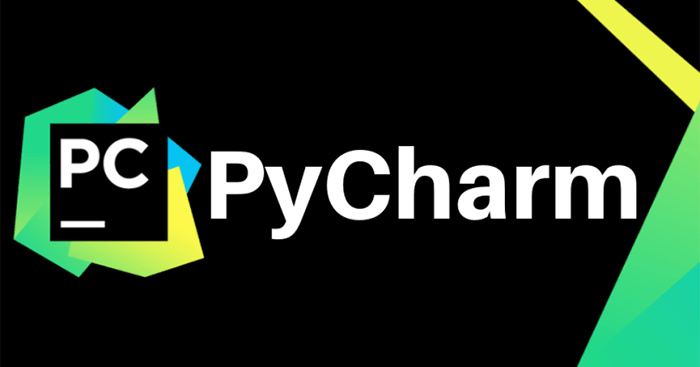
Hình 1.11. Biểu tượng android studio

Android Studio là môi trường phát triển tích hợp (IDE - Integrated Development Environment) được Google chính thức phát hành và hỗ trợ dành riêng cho việc phát triển ứng dụng Android. Với sự phát triển mạnh mẽ của Flutter – framework đa nền tảng của Google, Android Studio cũng cung cấp plugin để hỗ trợ lập trình Flutter hiệu quả.

**Tính năng nổi bật:**

* **Hỗ trợ đa nền tảng**: Dễ dàng phát triển ứng dụng cho Android, iOS (qua Flutter), máy tính để bàn và web.
* **Plugin Flutter & Dart**: Cung cấp giao diện thân thiện để tạo, chỉnh sửa và chạy ứng dụng Flutter trực tiếp trên trình giả lập hoặc thiết bị thật.
* **Code Editor mạnh mẽ**: Tích hợp các công cụ như tự động hoàn thành, kiểm tra lỗi cú pháp theo thời gian thực, và refactor code thông minh.
* **Android Emulator**: Cho phép mô phỏng thiết bị thật giúp dễ dàng kiểm thử ứng dụng mà không cần thiết bị vật lý.
* **Visual Layout Editor**: Với Flutter, có thể tận dụng các plugin hỗ trợ dựng UI bằng drag & drop hoặc theo mô hình Widget.
* **Tích hợp với Git và các hệ thống quản lý mã nguồn khác**, giúp việc làm việc nhóm trở nên dễ dàng và chuyên nghiệp hơn.

### Pycharm



Hình 1.12. Biểu tượng pycharm

**PyCharm** là một IDE nổi tiếng chuyên dụng cho ngôn ngữ lập trình Python, được phát triển bởi JetBrains – hãng phần mềm nổi tiếng với các công cụ phát triển như IntelliJ IDEA, WebStorm,... Với sự phát triển mạnh mẽ của Python trong lĩnh vực khoa học dữ liệu và trí tuệ nhân tạo, PyCharm trở thành công cụ không thể thiếu với các lập trình viên AI.

**Tính năng nổi bật:**

* **Hỗ trợ Python toàn diện**: Hệ thống gợi ý code, tự động sửa lỗi, định dạng lại mã nguồn và hỗ trợ debug nâng cao.
* **Quản lý môi trường ảo (virtual environment)**: Dễ dàng cô lập các thư viện và phiên bản Python để tránh xung đột giữa các dự án.
* **Tích hợp tốt với thư viện ML/AI**: Như TensorFlow, Keras, scikit-learn, NumPy, Pandas, giúp lập trình viên dễ dàng viết và huấn luyện mô hình.
* **Hỗ trợ Jupyter Notebook**: Cho phép tương tác với dữ liệu, trực quan hóa kết quả và thử nghiệm mô hình dễ dàng.
* **Tích hợp hệ thống version control**: Quản lý mã nguồn thông qua Git, GitHub hoặc GitLab.

Trong đồ án này, PyCharm được sử dụng để xây dựng và huấn luyện mô hình học sâu (Deep Learning) sử dụng thư viện TensorFlow. Mô hình CNN (Convolutional Neural Network) được huấn luyện để phân loại các loại hoa quả. Sau khi huấn luyện, mô hình sẽ được chuyển sang định dạng .tflite để triển khai trên thiết bị di động thông qua Flutter.

### TensorFlow

**TensorFlow** là một thư viện mã nguồn mở do Google phát triển và phát hành lần đầu tiên vào năm 2015. TensorFlow nhanh chóng trở thành một trong những nền tảng hàng đầu cho việc phát triển các ứng dụng học máy (Machine Learning - ML) và học sâu (Deep Learning - DL). TensorFlow hỗ trợ từ quá trình xây dựng mô hình, huấn luyện, đánh giá cho đến triển khai mô hình trên nhiều nền tảng khác nhau như web, máy chủ, thiết bị di động

**Tính năng nổi bật:**

* **Mô hình toán học linh hoạt**: Cho phép xây dựng các mô hình từ đơn giản đến phức tạp, như mạng nơ-ron tích chập (CNN), mạng tuần tự (RNN), attention model, transformer,...
* **Khả năng mở rộng cao**: Có thể huấn luyện mô hình trên CPU, GPU hoặc TPU, tăng tốc hiệu suất đáng kể.
* **TensorBoard**: Công cụ trực quan hóa giúp theo dõi quá trình huấn luyện như loss, accuracy, histogram, learning rate,...
* **Tích hợp dễ dàng với Keras**: Cho phép xây dựng mô hình với cú pháp thân thiện, dễ sử dụng.
* **Hỗ trợ TensorFlow Lite (TFLite)**: Giúp chuyển đổi mô hình sang định dạng nhẹ, tối ưu hóa để chạy trên các thiết bị di động hoặc nhúng.
* **Mô hình pretrained**: Có sẵn nhiều mô hình đã được huấn luyện trên các tập dữ liệu lớn, giúp tiết kiệm thời gian huấn luyện và cải thiện hiệu quả.

# CHƯƠNG 2. THIẾT KẾ HỆ THỐNG VÀ MÔI TRƯỜNG

## Cấu trúc mô hình MobileNetV2

## Chuẩn bị và phân tích dữ liệu

## Tiền xử lý dữ liệu

# CHƯƠNG 3. CÀI ĐẶT HỆ THỐNG VÀ MỘT SỐ KẾT QUẢ

## Huấn luyện mô hình

## Phân tích mô hình

# CHƯƠNG 4: TRIỂN KHAI ỨNG DỤNG DI ĐỘNG

## Các màn hình trong ứng dụng

## Thử nghiệm trên thiết bị thực tế

# KẾT LUẬN

# TÀI LIỆU THAM KHẢO

1. Lập trình hướng đối tượng Java – Nguyễn Văn Thành chủ biên.
2. Phạm Hữu Khang - Lập trình cơ sở dữ liệu tập II: thiết kế cơ sở dữ liệu, tạo các thủ tục xử lý.
3. Tham khảo tài liệu qua mạng Internet: trang web laptrinhjavaweb.com, youtube.com