TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

======\*\*\*======

BÁO CÁO THỰC NGHIỆM

HỌC PHẦN: HỆ HỖ TRỢ QUYẾT ĐỊNH

**Đề tài: Tìm hiểu về Deep Learning trong việc xây dựng hệ hỗ trợ**

**phân loại tiền Việt Nam**

|  |  |
| --- | --- |
| GVHD: | ThS. Trần Thanh Hùng |
| Nhóm: | 08 |
| Thành viên nhóm: | Lê Văn Phúc - 2021602585  Trần Thị Hoa Uyên - 2021602093  Nguyễn Thị Vân - 2021601412 |
| Lớp: 20232IT6058001 | Khoá:16 |

Hà Nội, 2023

**MỤC LỤC**

[LỜI NÓI ĐẦU 1](#_Toc169648057)

[CHƯƠNG 1: LÝ THUYẾT HỆ HỖ TRỢ RA QUYẾT ĐỊNH 2](#_Toc169648058)

[1.1. Hệ hỗ trợ ra quyết định là gì? 2](#_Toc169648059)

[1.1.1. DSS (Decision Support System) là gì? 2](#_Toc169648060)

[1.1.2. Khái niệm DSS (Decision Support System) 2](#_Toc169648061)

[1.1.3. Đặc điểm của hệ thống hỗ trợ ra quyết định DSS là gì? 2](#_Toc169648062)

[1.1.4. Lợi ích của DSS mang lại cho doanh nghiệp 3](#_Toc169648063)

[1.2. Kiến trúc chung hệ hỗ trợ ra quyết định 5](#_Toc169648064)

[1.3. Mô hình ra quyết định 6](#_Toc169648065)

[1.4. Mục đích xây dựng hệ hỗ trợ ra quyết định 7](#_Toc169648066)

[1.5. Tính cần thiết và phạm vi ứng dụng của hệ hỗ trợ ra quyết định 7](#_Toc169648067)

[1.5.1. Tính cần thiết của hệ hỗ trợ ra quyết định 7](#_Toc169648068)

[1.5.2. Phạm vi ứng dụng của hệ hỗ trợ ra quyết định 8](#_Toc169648069)

[1.6. Môi trường ra quyết định và các yếu tố ảnh hưởng 8](#_Toc169648070)

[1.6.1. Môi trường ra quyết định 8](#_Toc169648071)

[1.6.2. Các yếu tố ảnh hưởng đến ra quyết định 10](#_Toc169648072)

[CHƯƠNG 2: TÌM HIỂU VỀ DEEP LEARNING VÀ MÔ HÌNH CONVOLUTIONAL NEURAL NETWORK 12](#_Toc169648073)

[2.1. Deep Learning là gì? 12](#_Toc169648074)

[2.2. Deep Learning hoạt động như thế nào? 12](#_Toc169648075)

[2.3. Ưu và nhược điểm của Deep Learning 14](#_Toc169648076)

[2.4. Ứng dụng của Deep Learning vào đời sống 15](#_Toc169648077)

[2.4.1. Digital Marketing 15](#_Toc169648078)

[2.4.2. Chế tạo Robot 16](#_Toc169648079)

[2.4.3. Hệ thống lái xe tự động 17](#_Toc169648080)

[2.4.4. Trợ lý ảo 18](#_Toc169648081)

[2.4.5. Nghiên cứu y học 19](#_Toc169648082)

[2.4.6. Nhận diện khuôn mặt 20](#_Toc169648083)

[2.4.7. Mua sắm và giải trí 21](#_Toc169648084)

[2.5. Tổng quan về mạng CNN 22](#_Toc169648085)

[2.5.1. Khái niệm chung 22](#_Toc169648086)

[2.5.2. Các kiểu tầng trong mạng CNN 23](#_Toc169648087)

[2.5.3. Ví dụ minh họa mô hình CNN 28](#_Toc169648088)

[CHƯƠNG 3: XÂY DỰNG HỆ QUYẾT ĐỊNH SỬ DỤNG MÔ HÌNH CNN TRONG BÀI TOÁN NHẬN DIỆN TIỀN VIỆT NAM 31](#_Toc169648089)

[3.1. Phát biểu bài toán 31](#_Toc169648090)

[3.1.1. Phát biểu bài toán 31](#_Toc169648091)

[3.1.2. Thuật toán áp dụng 32](#_Toc169648092)

[3.2. Cơ sở dữ liệu 33](#_Toc169648093)

[3.3. Cài đặt 35](#_Toc169648094)

[3.3.1. Tiền xử lý dữ liệu 35](#_Toc169648095)

[3.3.2. Tiến hành huấn luyện 38](#_Toc169648096)

[3.3.3. Đánh giá và lưu mô hình 41](#_Toc169648097)

[3.3.4. Test mô hình 43](#_Toc169648098)

[TỔNG KẾT 46](#_Toc169648099)

[TÀI LIỆU THAM KHẢO 47](#_Toc169648100)

**DANH MỤC HÌNH ẢNH**

[Hình 1. 1: Sơ đồ các giai đoạn ra quyết định 5](#_Toc154081141)

[Hình 1. 2: Kiến trúc chung hệ hộ trợ quyết định 6](#_Toc154081142)

[Hình 1. 3 :Mô hình ra quyết định 7](#_Toc154081143)

[Hình 2. 1: Sơ đồ mô tả trước và sau phân loại bằng thuật toán KNN.................12](#_Toc154081153)

[Hình 2. 2: Bản đồ của 1NN (Nguồn: Wikipedia) 14](#_Toc154081154)

[Hình 2. 3:Ví dụ thuật toán KNN 15](#_Toc154081155)

[Hình 2. 4: Ví dụ về Kỹ thuật One-hot Encoding 17](#_Toc154081156)

[Hình 2. 5: Các công thức tính khoảng cách 18](#_Toc154081157)

[Hình 2. 6: K tăng dần làm min ranh giới 18](#_Toc154081158)

[Hình 2. 7: Đồ thị của K biến thiên trong K-Mean (Distortion) 19](#_Toc154081159)

[Hình 2. 8: Đồ thị của K biến thiên trong K-Mean (Inertia) 19](#_Toc154081160)

**DANH MỤC BẢNG**

[Bảng 1: Khái niệm người và máy 13](#_Toc154081429)

[Bảng 2: Bảng mẫu dữ liệu trong file CSV 25](#_Toc154081430)

# LỜI NÓI ĐẦU

Trong thời đại số hóa ngày nay, công nghệ thông tin và trí tuệ nhân tạo đang ngày càng chiếm giữ một vị trí quan trọng trong cuộc sống và các ngành công nghiệp. Đặc biệt, hệ thống hỗ trợ quyết định, với khả năng xử lý lượng thông tin lớn và tự động hóa quy trình ra quyết định, đang trở thành một công cụ không thể thiếu trong nhiều lĩnh vực.

Trong bối cảnh thị trường phim đa dạng với hàng ngàn thể loại, việc gợi ý phim dựa trên sở thích cá nhân của từng người xem trở nên quan trọng hơn bao giờ hết. Với mong muốn đáp ứng nhu cầu này, đề tài của chúng ta hôm nay đã chọn lựa mô hình K-Nearest Neighbors (KNN) - một phương pháp học máy cổ điển nhưng vẫn rất hiệu quả - để xây dựng một hệ thống gợi ý phim theo thể loại.

Mô hình KNN hoạt động dựa trên nguyên lý đơn giản nhưng mạnh mẽ: nếu một đối tượng có thuộc tính gì đó gần giống với các đối tượng trong tập dữ liệu đã biết, thì đối tượng đó cũng có khả năng thuộc vào lớp thể loại tương tự. Bằng cách này, chúng ta có thể dựa vào lịch sử xem phim của người dùng để đề xuất những bộ phim phù hợp và tăng trải nghiệm người dùng.

Trong bài trình bày tiếp theo, chúng ta sẽ cùng nhau tìm hiểu chi tiết về mô hình KNN, cách thức hoạt động, và cách áp dụng nó vào việc gợi ý phim theo thể loại. Hy vọng rằng, thông qua đề tài này, chúng ta sẽ có cái nhìn sâu rộng hơn về ứng dụng của trí tuệ nhân tạo trong cuộc sống và công việc hàng ngày.

***Một lần nữa, xin chân thành cảm ơn thầy***

# LÝ THUYẾT HỆ HỖ TRỢ RA QUYẾT ĐỊNH

## Hệ hỗ trợ ra quyết định là gì?

### DSS (Decision Support System) là gì?

DSS là viết tắt của cụm từ Decision Support System nghĩa là hệ thống hỗ trợ ra quyết định hoặc hệ hỗ trợ quyết định. Đúng như tên gọi của mình, DSS được tạo ra với mục đích hỗ trợ trong việc đưa ra quyết định cũng như đưa ra những dự đoán, chiều hướng hành động của một tổ chức hoặc một doanh nghiệp.

### Khái niệm DSS (Decision Support System)

DSS hay hệ hỗ trợ ra quyết định là một hệ thống phần mềm tương tác và thu thập mọi thông tin liên quan từ rất nhiều nguồn như vận hành, thu nhập, chi phí, thị trường, xu hướng, mô hình doanh nghiệp. Sau đó DSS sẽ sàng lọc và phân tích những dữ liệu này, tổng hợp lại thành các thông tin một cách toàn diện từ đó có thể sử dụng để hỗ trợ trong việc đưa ra quyết định.

DSS có thể được điều khiển thủ công bởi con người hoặc máy tính hóa hoàn toàn Trong một vài trường hợp, DSS có thể kết hợp cả hai phương pháp với nhau. Hệ thống DSS lý tưởng sẽ phân tích dữ liệu và tổng hợp thông tin để tăng độ chính xác cũng như tốc độ trong việc đưa ra quyết định.

### Đặc điểm của hệ thống hỗ trợ ra quyết định DSS là gì?

Hệ hỗ trợ ra quyết định - DSS được lập trình để tạo ra nhiều loại báo cáo khác nhau, tất cả đều tùy chỉnh được bởi thông số kỹ thuật của người dùng. Điều này là rất thuận tiện và hữu ích vì nó giúp việc trình bày những thông tin tới người xem trở nên dễ hiểu và phù hợp hơn. Thay vì một báo cáo đi qua nhiều mức quản lý như trước, với DSS có thể tạo ra các báo cáo khác nhau cho từng mức quản lý từ đó họ có thể đưa ra quyết định thích hợp tại mức riêng của họ.

[](https://marketingai.mediacdn.vn/wp-content/uploads/2019/10/DSS-app.jpg)

Hình 1.1: Đặc điểm của hệ thống hỗ trợ ra quyết định DSS là gì?

### Lợi ích của DSS mang lại cho doanh nghiệp

Sau khi hiểu DSS là gì? Có thể thấy lợi ích lớn nhất và hệ hỗ trợ ra quyết định - DSS mang lại là cung cấp những thông tin bổ ích cần thiết cho các nhà quản trị để từ đó họ đưa ra những quyết định chính xác hơn. Ngoài ra, DSS chỉ là một phần mềm vi tính nên người dùng có thể sử dụng vô cùng thuận tiện và linh hoạt. Cung cấp thông tin cho người dùng mọi lúc mọi nơi cũng như phù hợp với từng mức quản lý khác nhau với khả năng tạo ra nhiều loại báo cáo. Tất cả để phục vụ mục đích chát của DSS là cải thiện quy trình đưa ra quyết định giúp kiểm soát doanh nghiệp một cách hiệu quả hơn.

#### Vai trò hệ hỗ trợ ra quyết định

***Lý do sử dụng DSS:***

* Cải thiện tốc độ tính toán;
* Tăng năng suất của cá nhân liên đới;
* Cải tiến kỹ thuật trong việc lưu trữ, tìm kiếm, trao đổi dữ liệu trong và ngoài tổ chức theo hướng nhanh và kinh tế;
* Nâng cao chất lượng của các quyết định đưa ra;
* Tăng cường năng lực cạnh tranh của tổ chức;
* Khắc phục khả năng hạn chế của con người trong việc xử lý và lưu chứa thông tin.

***Các hỗ trợ từ DSS:***

* Cung cấp thông tin trạng thái và dl thô
* Khả năng phân tích tổng quát
* Mô hình biểu diễn (cân đối tài chính), mô hình nhân quả (dự báo, chuẩn đoán)
* Đề nghị giải pháp, đánh giá
* Chọn lựa giải pháp

#### Quá trình ra quyết định của hệ hỗ trợ ra quyết định

1. **Phân loại quyết định**

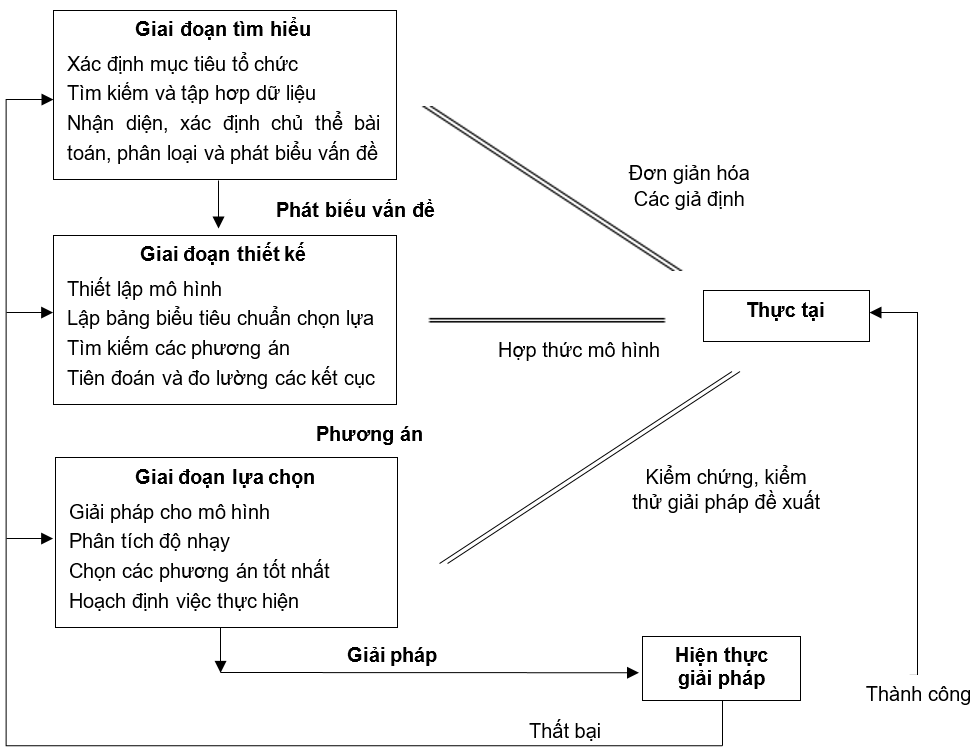
***Có thể phân ra 4 loại quyết định sau:***

* Quyết định có cấu trúc (Structured Decision): các quyết định mà người ra quyết định biết chắc chắn đúng.
* Quyết định không có cấu trúc (Nonstructured Decision): các quyết định mà người ra quyết định biết là có nhiều câu trả lời gần đúng và không có cách nào tìm ra câu trả lời chính xác nhất.
* Quyết định đệ quy (Recurring Decision): các quyết định lặp đi, lặp lại
* Quyết định không đệ quy (Nonrecurring Decision): các quyết định không xảy ra thường xuyên.

1. **Các giai đoạn của quá trình ra quyết định**

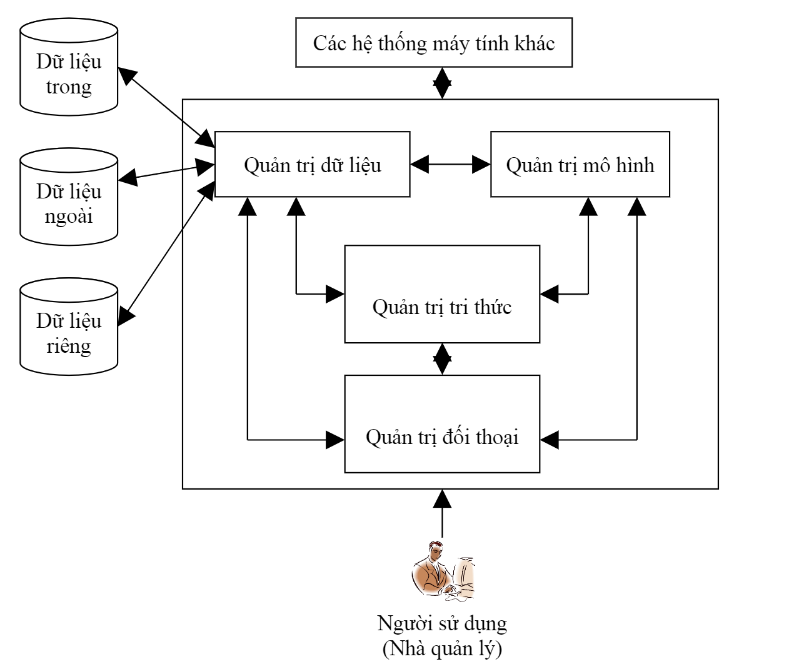
Theo Simon, các giai đoạn của quá trình ra quyết định gồm các pha:

* Tìm hiểu - bài toán dẫn đến quyết định;
* Thiết kế - phân tích và xây dựng các diễn trình hành động;
* Chọn lựa - chọn một diễn trình trong tập diễn trình;
* Thực hiện - thực hiện các quyết định để có được kết quả.



Hình 1. 1: Sơ đồ các giai đoạn ra quyết định

## Kiến trúc chung hệ hỗ trợ ra quyết định



Hình 1. 2: Kiến trúc chung hệ hộ trợ quyết định

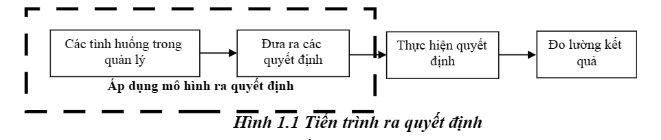
Trong đó:

* Quản trị dữ liệu: bao gồm các CSDL chứa dữ liệu liên quan đến một tình huống và được quản lý bởi phần mềm là hệ quản trị CSDL (quản lý và khai thác).
* Quản trị mô hình: cho phép khai thác và quản lý các mô hình định lượng (xử lý) khác nhau, cung cấp khả năng phân tích cho hệ thống.
* Quản trị đối thoại: cung cấp giao diện cho người dùng để liên lạc và ra lệnh cho Hệ hỗ trợ quyết định.
* Quản trị tri thức: hoạt động như 1 thành phần độc lập, hoặc có thể trợ giúp cho bất kỳ 1 hệ thống nào trong 3 hệ thống nói trên.

## Mô hình ra quyết định

Mô hình ra quyết định là một lĩnh vực khoa học quản trị nhằm tìm ra phương pháp tối ưu hoặc hiệu quả nhất của việc sử dụng các nguồn lực có hạn để có thể đạt được các mục tiêu của một cá nhân hoặc một doanh nhiệp đưa ra. Vì lý do này, mô hình ra quyết định thường được hiểu với một nghĩa khác là *Tối ưu hóa.*

Mô hình ra quyết định thường chỉ áp dụng hai giai đoạn đầu của tiến trình ra quyết định đó là các tình huống quản lý và đưa ra các quyết định còn lại các bước thực hiện quyết định và đo lường kết quả đạt được khi ra quyết định thì không được đề cập đến.



Hình 1. 3 :Mô hình ra quyết định

Triết lý về ra quyết định mỗi cách tiếp cận đối với các vấn đề khác nhau, phụ thuộc rất nhiều vào môi trường xung quanh, nền tảng kiến thức, kinh nghiệm, trạng thái tâm lý.

***Có nhiều mô hình ra quyết định, nổi bật nhất là:***

* Mô hình tỷ lệ: xem ra quyết định là một quá trình có cấu trúc, rút gọn bài toán thành lập các tham số đo được.
* Mô hình tổ chức: quan tâm nhiều đến các chính sách, định hướng tiếp theo, quan tâm nhiều tới việc định tính.
* Mô hình chính trị: kết quả của liên kết nhóm, thể hiện các khả năng cá nhân.

## Mục đích xây dựng hệ hỗ trợ ra quyết định

Mục đích chính của việc sử dụng DSS là trình bày thông tin cho khách hàng một cách dễ hiểu. Một DSS rất hữu ích vì nó có thể được lập trình để tạo nhiều loại báo cáo, tất cả dựa trên thông số kĩ thuật của người dùng.

DSS trợ giúp các hoạt động ra quyết định. Hệ thống này có rất nhiều ưu điểm nổi trội, cung cấp cho người dùng những thông tin hữu ích một cách linh hoạt.

Cung cấp các công cụ trợ giúp việc phát triển và cải thiện các mô hình nhận thức (về nhân và quả) của người ra quyết định bằng cách cung cấp dữ liệu nhanh, đúng & áp dụng các mô hình toán học.

## Tính cần thiết và phạm vi ứng dụng của hệ hỗ trợ ra quyết định

### Tính cần thiết của hệ hỗ trợ ra quyết định

* Nghiên cứu và hoạch định tiếp thị: Chính sách giá cho kháchhàng, dự báo sản phẩm tiêu thụ…
* Hoạch định chiến lược và vận hành: Theo dõi, phân tích và báo cáo về xu hướng thị trường…
* Hỗ trợ bán hàng: Chi tiết và tổng hợp tình hình bán hàng, so sánh và phân tích xu hướng bán hàng…

### Phạm vi ứng dụng của hệ hỗ trợ ra quyết định

***Ở đâu:***

* Ở các quyết định hàng ngày.
* Các quyết định có tính chiến thuật:chọn 1 phương sách thích hợp để đáp ứng 1 mục tiêu nào đó như chuẩn bị ngân sách, quản lý tài chính.
* Các quyết định có tính chiến lược:quyết định chính sách dài hạn, đầu tư dài hạn, tổ chức lại cơ quan, chiến lược tiếp thị.

***Khi nào:***

* Tài chính: mua trang thiết bị, phần mềm.
* Tổ chức: thích hợp.
* Hiệu quả và rủi ro: giảm rủi ro, không phải là hoàn vốn

***Các loại bài toán áp dụng:***

* Đơn giản.
* Có nhiều phương án chọn hơn, hàm đánh giá đơn giản.
* Phức tạp: đa chỉ tiêu.
* Rất phức tạp: vượt quá khả năng xử lý của con người.

***Như thế nào:***

* Thân thiện với người sử dụng về giao diện.
* Tính cập nhật thường xuyên cập nhật trong môi trường biến động.
* Độ chi tiết đảm bảo yêu cầu người sử dụng. Tần suất sử dụng cao, tính thuận tiện: mềm dẻo, thiết kế tốt

## Môi trường ra quyết định và các yếu tố ảnh hưởng

### Môi trường ra quyết định

***Các yếu tố tác động***: trực tiếp dễ nhìn ra, dễ lượng hoá, dễ nhìn, dễ hiểu, dễ cấu trúc hoá.

***Các yếu tố về tổ chức tác động đến mọi bài toán quyết định:***

* Chính sách: luật, mệnh lệnh, quan hệ, vay, trả.
* Cấu trúc tổ chức: cách quản lý, cách điều hành, quy chế.
* Uy tín của tổ chức có thể lượng hoá.
* Con người trong tổ chức: hành vi văn hoá, thái độ nhân cách.

***Các yếu tố ngoại cảnh:***

* Tình hình kinh tế.
* Tình hình thị trường.
* Tình hình môi trường.
* Pháp luật: chính sách có thể thay đổi còn pháp luật khó thay đổi.
* Sự chấp nhận của khách hàng.

***Các yếu tố thông tin:***

* Khả năng liên lạc.
* Độ bảo mật của thông tin.
* Độ tin cậy của thông tin: sự chính xác, mức độ cập nhật của thông tin.
* Các giải pháp thông tin tổng hợp, đa dạng, tỉ mỉ chi tiết.
* Giá cả của thông tin: thu nhập, truyền, xử lý.

***Các mục tiêu về quản lý:***

* Sự vận hành: kinh tế, hiệu quả, chất lượng, an toàn.
* Định lượng hay định tính.
* Mức độ rõ ràng của môi trường ra quyết định: rõ, ẩn hay mờ.

***Phân loại môi trưởng ra quyết định:***

* Môi trường truy nhập được và không truy nhập được.
* Môi trường tất định, không tất định.
* Môi trường tĩnh, động.
* Môi trường chắc chắn, không chắc chắn.

### Các yếu tố ảnh hưởng đến ra quyết định

***Trực tiếp, dễ nhìn thấy, dễ hiểu như là:***

* Hạn chế về tài nguyên (ràng buộc ngân sách, khả năng phát triển sản xuất,...)
* Điều kiện vật lý (các tham số đo, các khoảng cách liên hệ,...)
* Các tham số chức năng ảnh hưởng đến hiệu quả (tỷ lệ sản xuất, độ tin cậy của nhà cung cấp, chất lượng sản phẩm, tay nghề nhân công, tỷ lệ sai hỏng, chỉ số kinh tế,...

***Các yếu tố tổ chức:***

* Chính sách: các luật mệnh lệnh, các quan hệ, sự vay trả, sự định hướng thực hiện.
* Cấu trúc: vị trí địa lý, cách quản lý, nghiên cứu – phát triển, cách điều hành (tập trung, phân cấp, ma trận).
* Hình ảnh (uy tín): uy tín xã hội (công cộng), uy tín kinh doanh (thỏa mãn khách hàng, đảm bảo tài chính), uy tín trong nội bộ (không mâu thuẫn).
* Con người: điều kiện xã hội, ý thức chấp hành, hành vi văn hóa, thái độ, nhân cách.

***Các yếu tố ngoại cảnh:***

* Pháp luật, các quy định cần tuân thủ (thời gian sản xuất, môi trường sản xuất, luật chất thải, giới hạn giá cả, xuất khẩu,...)
* Kinh tế: sự đầu tư tài chính, giá cả, thuế.
* Môi trường: thời tiết, yếu tố địa lý, thiên tai.
* Thị trường: sự cạnh tranh, phát triển công nghệ mới.
* Đòi hỏi của khách hàng, nhu cầu của khách hàng.

***Yếu tố thông tin:***

* Khả năng thông tin: độ bảo mật, khả năng truyền thông, phương tiện lưu trữ, nguồn, kênh thông tin (bên trong, bên ngoài).
* Độ tin cậy: sự chính xác, cập nhật, chuẩn mực.
* Giải pháp: thông tin tỉ mỉ, tổng hợp, đa dạng.
* Giá cả: cho thu thập, chuẩn bị, kiểm tra, bảo hành thông tin, cả về thời gian tính và đòi hỏi phần cứng.

# TÌM HIỂU VỀ DEEP LEARNING VÀ MÔ HÌNH CONVOLUTIONAL NEURAL NETWORK

## Deep Learning là gì?

Deep Learning là một nhánh của Machine Learning và Khoa học dữ liệu bắt chước cách con người thu được những kiến thức cụ thể, giúp máy tính tự huấn luyện chính nó thực hiện những tác vụ tương tự con người, hỗ trợ cho việc dịch ngôn ngữ, phân loại các hình ảnh, nhận dạng giọng nói.

Các thuật toán Deep Learning là nhân tố quan trọng đằng sau những chiếc xe không người lái, cho phép chúng nhận ra biển báo dừng hoặc phân biệt người đi bộ với cột đèn. Công nghệ này là chìa khóa để con người có thể dùng giọng nói điều khiển các thiết bị tiêu dùng như điện thoại, máy tính bảng, TV và loa cầm tay. Qua từng bước phát triển, Deep Learning đang đạt được những kết quả chưa từng có trước đây.



Hình 2.1: Ô tô không người lái ứng dụng Deep Learning

## Deep Learning hoạt động như thế nào?

Sau khi hiểu về định nghĩa **Deep Learning là gì** thì chúng ta cần hiểu cách thức nó vận hành ra sao để có cái nhìn tổng quát nhất.

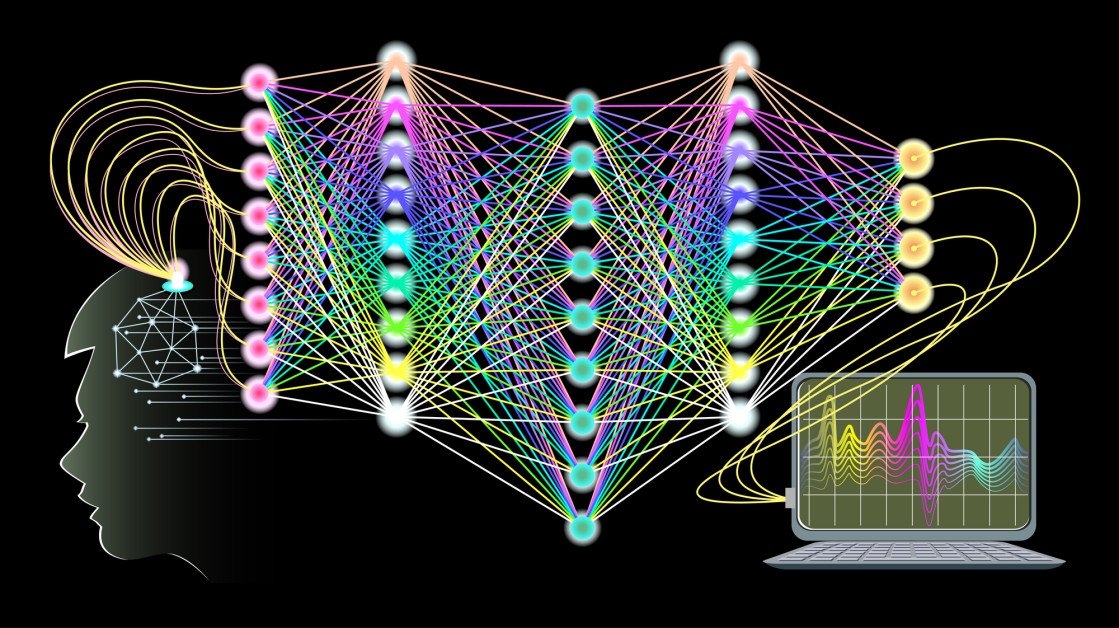
Deep Learning về cơ bản là một mạng nơ-ron gồm ba lớp trở lên, những mạng lưới nơ-ron này thu thập từ dữ liệu lớn (Big Data), học từ những dữ liệu đó và cố gắng “bắt chước” hành vi của não bộ con người. Công nghệ này hoạt động bằng cách khám phá các cấu trúc phức tạp trong dữ liệu mà chúng trải nghiệm, cụ thể là xây dựng các mô hình tính toán bao gồm nhiều lớp xử lý, mạng có thể tạo ra nhiều mức trừu tượng để biểu diễn dữ liệu.

Học sâu hoạt động bằng cách khám phá các cấu trúc phức tạp trong dữ liệu mà chúng thu thập. Các giai đoạn quá trình làm việc của Deep Learning như sau:

* ANN đặt ra một loạt các truy vấn đúng/sai nhị phân.
* Trích xuất các số từ các khối dữ liệu.
* Sắp xếp dữ liệu thành các danh mục dựa trên các phản hồi.
* Ghi nhãn dữ liệu.

Deep Learning rút ra kết luận và gắn nhãn dữ liệu mới, chưa được khám phá trong giai đoạn suy luận bằng cách sử dụng kiến ​​thức và kinh nghiệm trước đó của nó.

Ví dụ: Một mô hình Deep Learning được gọi là mạng nơ-ron tích chập có thể được tạo ra bằng cách sử dụng số lượng lớn (lên đến hàng triệu) hình ảnh và mạng này học hỏi từ các pixel có trong hình ảnh mà nó thu được. Nó có thể phân loại các nhóm pixel đại diện cho các đặc điểm của mèo, với các nhóm đặc điểm như móng vuốt, tai và mắt cho biết sự hiện diện của mèo trong hình ảnh.



Hình 2.2: Minh họa mạng Neural network

## Ưu và nhược điểm của Deep Learning

Một số ưu và nhược điểm của Deep Learning có thể kể đến như sau:

|  |  |
| --- | --- |
| **Ưu điểm** | **Nhược điểm** |
| * Deep Learning được tối ưu tốt hơn nhờ sự bùng nổ của Internet và khả năng phát triển mạnh mẽ từ khả năng tính toán của máy tính. * Độ chính xác cao, đảm bảo cho các thiết bị điện tử tiêu dùng có thể đáp ứng được mọi nhu cầu cũng như mong đợi của người dùng. * Cung cấp hệ thống gợi ý trên các nền tảng lớn như: Facebook, Amazon, Netflix,…, giúp tăng độ tương tác của người dùng lên cao đáng kể. * Khả năng nhận diện hình ảnh tốt, không chỉ xác định được đối tượng trong ảnh mà còn hiểu được nội dung và ngữ cảnh trong bức hình. * Độ an toàn cao, được tin dùng trong hệ thống lái xe tự động. | * Đòi hỏi một nguồn dữ liệu đầu vào khủng để máy tính học hỏi. Quy trình này mất nhiều thời gian và công sức để xử lý mà chỉ có các Server chuyên nghiệp mới có thể làm được. Trường hợp không đủ dữ liệu đầu vào để xử lý, mọi thứ sẽ không thể diễn ra đúng như dự định của máy tính đưa ra. * Chưa thể nhận biết được những thứ phức tạp. Bởi, kỹ thuật của Deep Learning hiện chưa đảm bảo tốt để trí tuệ nhân tạo có thể rút ra những kết luận một cách logic. |

## Ứng dụng của Deep Learning vào đời sống

Học sâu là một thuật ngữ tương đối xa lạ với những người chưa từng tìm hiểu về chủ đề này tuy nhiên Deep Learning đã được ứng dụng vào rất nhiều khía cạnh cũng như lĩnh vực trong đời sống của chúng ta, giúp ích rất nhiều cho các nhà khoa học dữ liệu trong việc thu thập, giải thích và phân tích khối lượng dữ liệu khổng lồ một cách dễ dàng và nhanh chóng. Deep Learning đã mang lại nhiều ứng dụng tích cực và đa dạng trong đời sống hàng ngày, từ nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên, đến tự động lái xe và chăm sóc sức khỏe. Các ứng dụng này không chỉ cải thiện hiệu suất và tiện ích của nhiều lĩnh vực, mà còn mở ra những triển vọng mới trong việc giải quyết các thách thức xã hội và kinh tế. Đồng thời, Deep Learning đang chứng minh vai trò quan trọng trong việc thúc đẩy sự phát triển và tiến bộ trong các lĩnh vực quan trọng của xã hội. Dưới đây là một số những ví dụ cụ thể về hình thức mà Deep Learning hiện diện và giúp ích cho cuộc sống của con người.

### Digital Marketing

Thời đại chuyển đổi số trên khắp mọi lĩnh vực hiện nay khiến việc [ứng dụng trí tuệ nhân tạo trong Digital Marketing](https://vtc.edu.vn/mot-so-ung-dung-dien-hinh-cua-tri-tue-nhan-tao-trong-digital-marketing) không còn quá xa lạ đối với dân kinh doanh lẫn dân công nghệ. Việc sử dụng Deep Learning trong chiến dịch Marketing sẽ giúp mang lại trải nghiệm tốt hơn cho khách hàng và tăng doanh thu cho doanh nghiệp. Một số ví dụ điển hình như [lợi ích của hệ thống Chatbot](https://vtc.edu.vn/5-loi-ich-khong-the-bo-qua-cua-chatbot-doi-voi-doanh-nghiep), xây dựng nội dung cá nhân hóa, phân tích người dùng bằng cách nhận diện hình ảnh, …

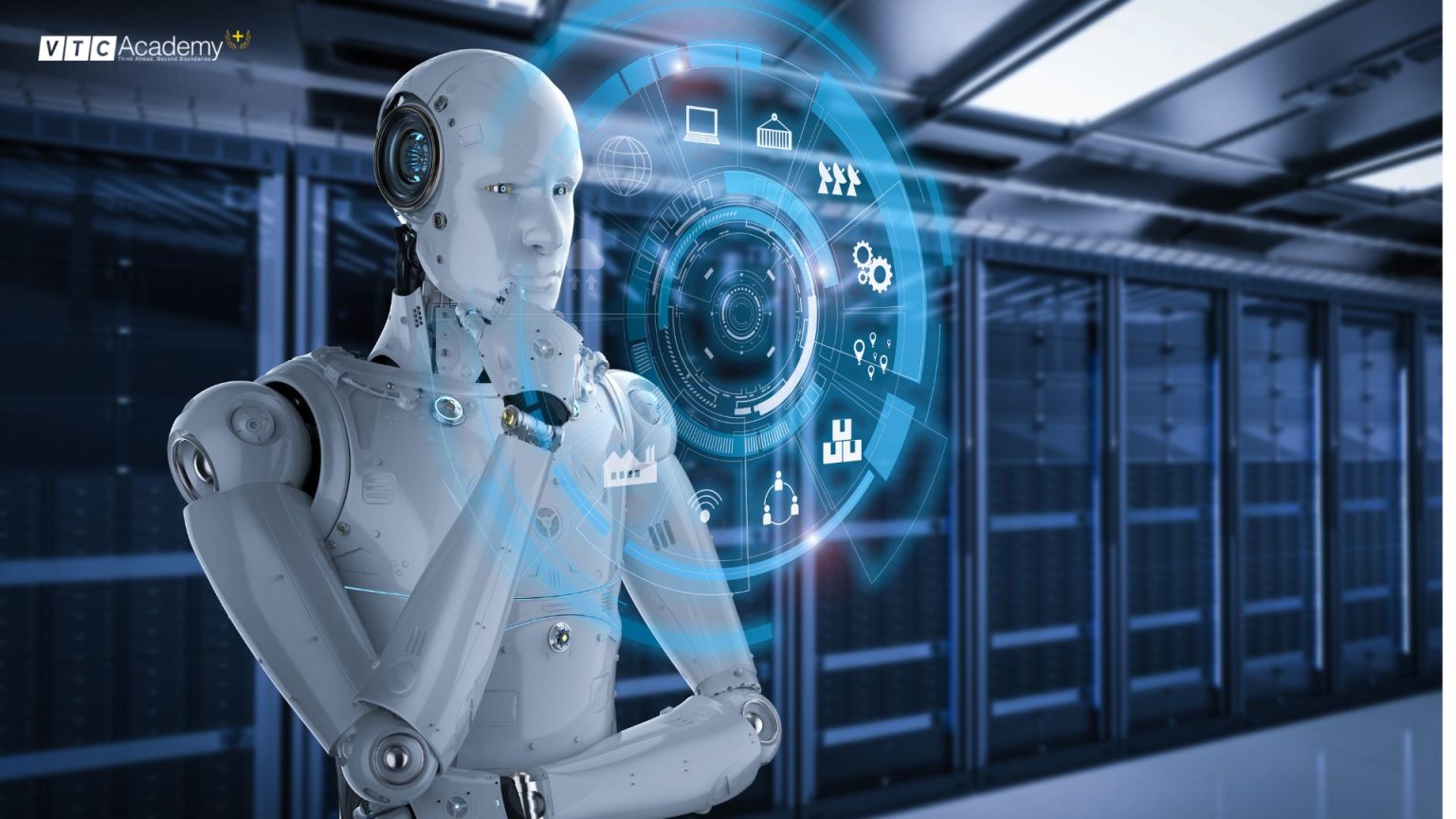


Hình 2.3: Ứng dụng Deep Learning vào Digital Marketing giúp chiến lược Marketing thành công hơn

### Chế tạo Robot

Nhiều sự phát triển gần đây trong chế tạo robot đã được thúc đẩy bởi những tiến bộ trong AI và Deep Learning. Ví dụ, AI cho phép robot cảm nhận và phản ứng với môi trường quanh nó, điều hướng, phân loại và xử lý các đồ vật không bằng phẳng, dễ vỡ hoặc sắp xếp những vật lộn xộn lại với nhau.

Sự phát triển của AI có nghĩa là chúng ta có thể mong đợi các robot trong tương lai sẽ ngày càng được sử dụng làm trợ lý cho con người. Chúng sẽ không chỉ được sử dụng để hiểu và trả lời các câu hỏi, như một số được sử dụng ngày nay. Người dùng kỳ vọng chúng sẽ có thể thực hiện các lệnh bằng giọng nói và cử chỉ, thậm chí dự đoán hành động tiếp theo của con người. Ngày nay, rô-bốt đã hoạt động cộng tác cùng với con người, trong đó con người và robot thực hiện các nhiệm vụ riêng biệt phù hợp nhất với thế mạnh của mỗi bên .



Hình 2.4: Lĩnh vực chế tạo Robot được dự đoán sẽ phát triển hơn trong tương lai nhờ sự bùng nổ của AI

### Hệ thống lái xe tự động

Một trong những sản phẩm áp dụng Deep Learning mới và hiện đại nhất hiện nay là hệ thống lái xe tự động, được xây dựng dựa trên các mạng nơ-ron cấp cao.

Các mô hình Deep Learning đóng vai trò nhận diện các đối tượng trong môi trường xung quanh xe, tính toán khoảng cách giữa xe và các phương tiện khác, xác định vị trí làn đường, tín hiệu giao thông, nhận diện chướng ngại vật,… từ đó đưa ra được các quyết định tối ưu và nhanh chóng nhất. Các thuật toán Học sâu càng nhận được nhiều dữ liệu thì càng có khả năng hành động giống như con người trong quá trình xử lý thông tin — phân biệt được biển báo dừng dù bị phủ tuyết …

Một trong những hãng xe đình đám và tiên phong trong việc sản xuất xe tự lái hiện nay là Tesla.



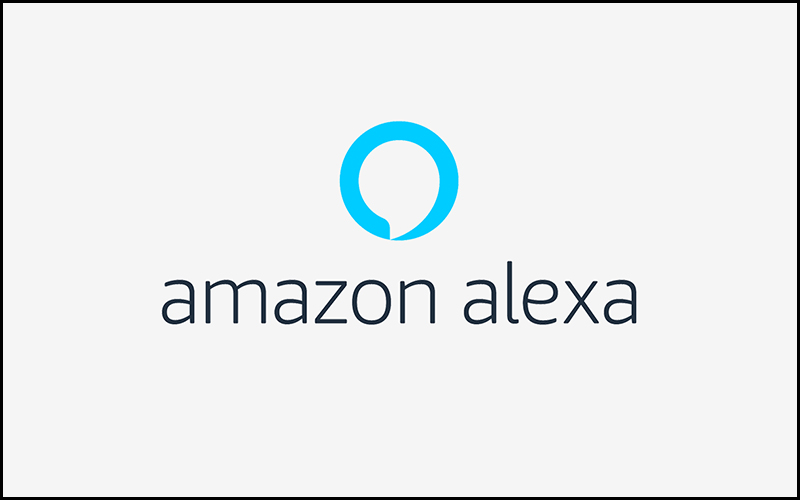
Hình 2.5: Tesla – hãng xe lái tự động nổi tiếng nhất hiện nay

### Trợ lý ảo

Trợ lý ảo không còn quá xa lạ với công chúng trong những năm gần đây bởi những hiệu quả vượt trội mà chúng mang lại.  
Trợ lý ảo hay Virtual Assistant đang được ứng dụng rất nhiều trong đời sống hàng ngày, phổ biến nhất nhất có thể kể đến Alexa hay Siri, Google Assistant. Mỗi tương tác với các trợ lý này cung cấp cho họ cơ hội tìm hiểu thêm về giọng nói và ngữ điệu của bạn, từ đó cung cấp cho bạn trải nghiệm tương tác như phiên bản thứ 2 của con người.

Trợ lý ảo sử dụng Deep Learning để biết thêm về người dùng, từ sở thích ăn uống đến các điểm truy cập nhiều nhất hoặc các bài hát yêu thích của bạn. Công nghệ này học cách hiểu và thực hiện các mệnh lệnh thông qua đánh giá ngôn ngữ tự nhiên của con người.

Ngoài ra Deep Learning còn có khả năng giúp trợ lý ảo dịch bài phát biểu thành văn bản, ghi chú lại và đặt lịch hẹn cho bạn; nhắc lịch hẹn và tự động trả lời các cuộc gọi cụ thể để phối hợp các nhiệm vụ giữa bạn và các thành viên chung team; tạo hoặc gửi bản sao email phù hợp.



Hình 2.6: Alexa – trợ lý ảo thông minh được phát triển bởi Amazon

### Nghiên cứu y học

Từ chẩn đoán bệnh, chẩn đoán khối u cho đến các loại thuốc được tạo riêng cho từng bộ gen cá nhân, ứng dụng của Deep Learning vào y học đang nhận được sự chú ý từ nhiều công ty dược phẩm và y tế lớn nhất hiện nay.

Các nhà nghiên cứu ung thư đang sử dụng Deep Learning để tự động phát hiện tế bào ung thư. Các nhóm nghiên cứu tại UCLA đã chế tạo một kính hiển vi tiên tiến mang lại bộ dữ liệu đa chiều được sử dụng để đào tạo một ứng dụng Deep Learning nhằm xác định chính xác các tế bào ung thư.

Số hóa hồ sơ bệnh nhân và đọc hình ảnh cũng là minh chứng nổi bật của việc ứng dụng công nghệ này vào trong ngành y tế. Deep Learning phân tích tự động hình ảnh chụp cắt lớp của bệnh nhân, giúp bác sĩ chẩn đoán chính xác và tình trạng bệnh được phát hiện nhanh chóng, có biện pháp điều trị từ sớm. Quá trình kiểm tra sức khỏe cũng trở nên đơn giản khi mọi dữ liệu của người bệnh đã được ghi nhận trên hệ thống chung.



Hình 2.7: Deep Learning ứng dụng rộng rãi trong nghiên cứu y học

### Nhận diện khuôn mặt

Deep Learning được sử dụng trong nhận diện khuôn mặt không chỉ nhằm phục vụ cho mục đích bảo mật mà còn được ứng dụng việc phân loại hình ảnh. Chắc hẳn, chúng ta đều đã từng thấy máy tính tự động nhận diện và phân loại các hình ảnh trên điện thoại.

Ví dụ: Facebook có thể tự động gắn thẻ chính bạn và bạn bè của bạn, hoặc Google Photos có thể tự động gắn nhãn ảnh của bạn để tìm kiếm dễ dàng hơn. Với Deep Learning, bạn có thể dễ dàng tìm và phân loại các hình ảnh theo ngày, sự kiện mà không phải dùng thao tác thủ công mất thời gian. Và trong tương lai gần, người dùng còn có thể thanh toán các hóa đơn bằng chính khuôn mặt của mình.

Tuy nhiên, thách thức đặt ra đối với Deep Learning khi nhận diện khuôn mặt là làm thế nào để xác định được chính xác một người ngay cả khi người đó thay đổi kiểu tóc, để râu hay cạo râu, hoặc khi chất lượng ảnh kém do ánh sáng hoặc môi trường xung quanh…

**

Hình 2.8: Deep Learning được sử dụng trong Face ID

### Mua sắm và giải trí

Bạn đã bao giờ thắc mắc làm thế nào Tiktok đưa ra các đề xuất về nội dung bạn nên xem tiếp theo chưa? Hoặc Shopee đưa ra ý tưởng về những gì bạn nên mua tiếp theo và những gợi ý đó chính xác là những gì bạn cần nhưng chưa từng biết trước đây? Đúng, đó là cách mà các thuật toán Deep Learning đang hoạt động, len lỏi vào trong các hoạt động giải trí của chúng ta. Deep Learning càng “học” được từ nhiều dữ liệu, chúng càng trở nên tốt hơn, từ đó những gợi ý đưa ra cũng chính xác hơn.

**

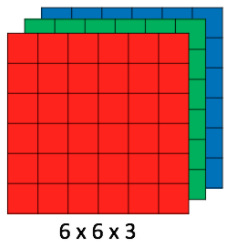
Hình 2.9: Netflix sử dụng Deep Learning để gợi ý những bộ phim phù hợp với sở thích người xem

## Tổng quan về mạng CNN

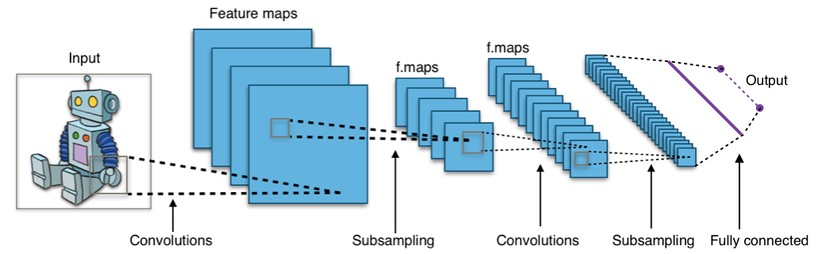
### Khái niệm chung

- Trong mạng neural, mô hình mạng neural tích chập (CNN) là 1 trong những mô hình để nhận dạng và phân loại hình ảnh. Trong đó, xác định đối tượng và nhận dạng khuôn mặt là 1 trong số những lĩnh vực mà CNN được sử dụng rộng rãi.

- CNN phân loại hình ảnh bằng cách lấy 1 hình ảnh đầu vào, xử lý và phân loại nó theo các hạng mục nhất định (Ví dụ: Chó, Mèo, Hổ, ...). Máy tính coi hình ảnh đầu vào là 1 mảng pixel và nó phụ thuộc vào độ phân giải của hình ảnh. Dựa trên độ phân giải hình ảnh, máy tính sẽ thấy H x W x D (H: Chiều cao, W: Chiều rộng, D: Độ dày).



Về kỹ thuật, mô hình CNN để training và kiểm tra, mỗi hình ảnh đầu vào sẽ chuyển nó qua 1 loạt các lớp tích chập với các bộ lọc (Kernals), tổng hợp lại các lớp được kết nối đầy đủ (Full Connected) và áp dụng hàm Softmax để phân loại đối tượng có giá trị xác suất giữa 0 và 1.



### Các kiểu tầng trong mạng CNN

##### a, Convolution Layer

Convolution (lớp tích chập) là lớp đầu tiên trích xuất các đặc tính từ hình ảnh. Tham số lớp này bao gồm một tập hợp các bộ lọc có thể học được. Các bộ lọc đều nhỏ thường có kích cỡ hai chiều đầu tiên khoảng 3x3 hoặc 5x5, .... và có độ sâu bằng với độ sâu của đầu vào đầu vào. Bằng cách trượt dần bộ lọc theo chiều ngang và dọc trên ảnh, chúng thu được một Feature Map chứa các đặc trưng được trích xuất từ trên hình ảnh đầu vào.

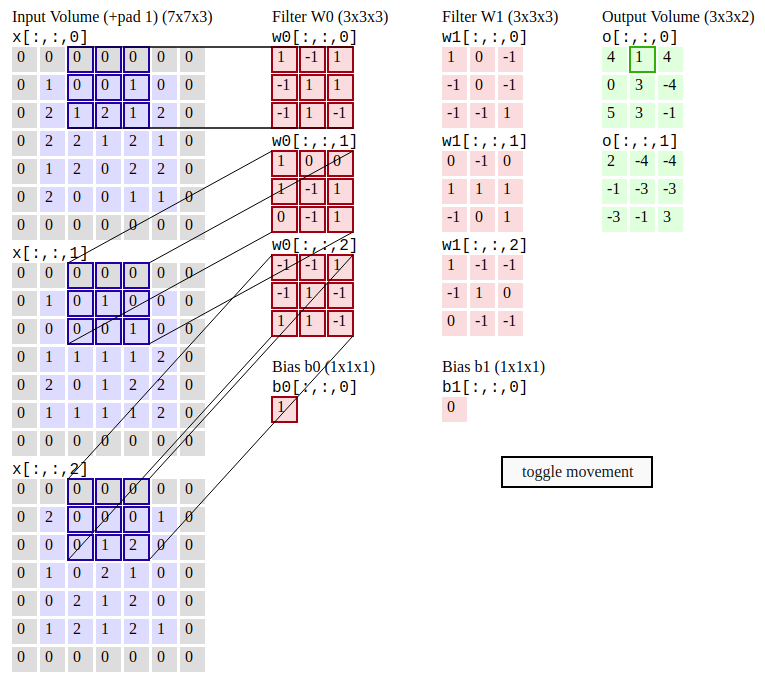
Quá trình trượt các bộ lọc thường có các giá trị được quy định bao gồm:

* Bước nhảy – Stride: Stride là số pixel thay đổi trên ma trận đầu vào. Khi stride là 1 thì ta di chuyển các kernel 1 pixel. Khi stride là 2 thì ta di chuyển các kernel đi 2 pixel và tiếp tục như vậy. Hình dưới là lớp tích chập hoạt động với stride là 2.
* Đường viền – Padding: Đôi khi kernel không phù hợp với hình ảnh đầu vào. Ta có 2 lựa chọn:
* Chèn thêm các số 0 vào 4 đường biên của hình ảnh (padding).
* Cắt bớt hình ảnh tại những điểm không phù hợp với kernel.
* Hàm phi tuyến – ReLU:

ReLU viết tắt của Rectified Linear Unit, là 1 hàm phi tuyến. Với đầu ra là: ƒ (x) = max (0, x).

Tại sao ReLU lại quan trọng: ReLU giới thiệu tính phi tuyến trong ConvNet. Vì dữ liệu trong thế giới mà chúng ta tìm hiểu là các giá trị tuyến tính không âm.

Có 1 số hà phi tuyến khác như tanh, sigmoid cũng có thể được sử dụng thay cho ReLU. Hầu hết người ta thường dùng ReLU vì nó có hiệu suất tốt.



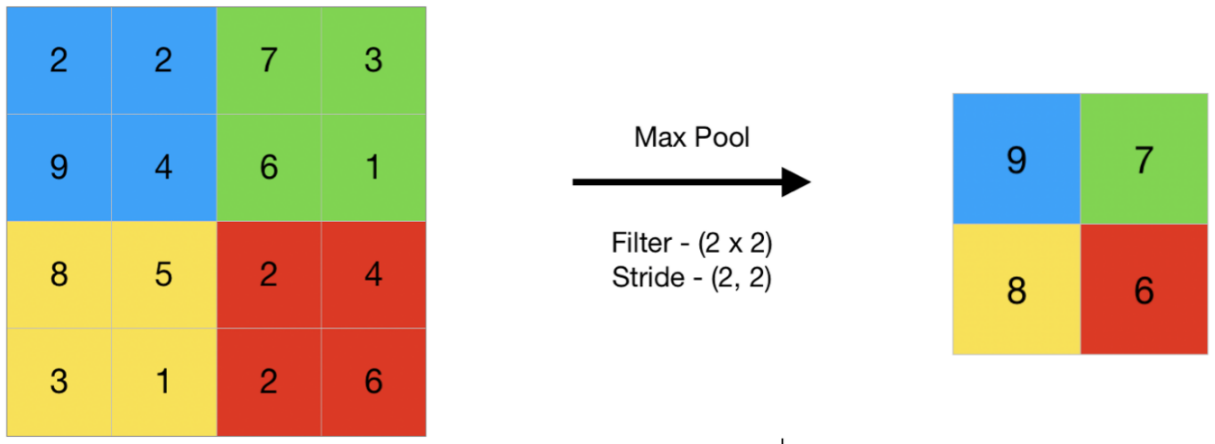
Với mỗi kernel khác nhau ta sẽ học được những đặc trưng khác nhau của ảnh, nên trong mỗi convolutional layer ta sẽ dùng nhiều kernel để học được nhiều thuộc tính của ảnh. Vì mỗi kernel cho ra output là 1 matrix nên k kernel sẽ cho ra k output matrix. Ta kết hợp k output matrix này lại thành 1 tensor 3 chiều có chiều sâu k. Output của convolutional layer sẽ qua hàm activation function trước khi trở thành input của convolutional layer tiếp theo.

##### b, Pooling layer

Mục tiêu của lớp gộp là lấy các tính năng quan trọng nhất từ ​​ma trận phức tạp. Điều này được thực hiện bằng cách áp dụng một số thao tác tổng hợp, giúp giảm kích thước của bản đồ đặc trưng (ma trận phức tạp), do đó giảm bộ nhớ được sử dụng trong khi huấn luyện mạng. Pooling cũng có liên quan để giảm thiểu tình trạng trang bị quá mức.

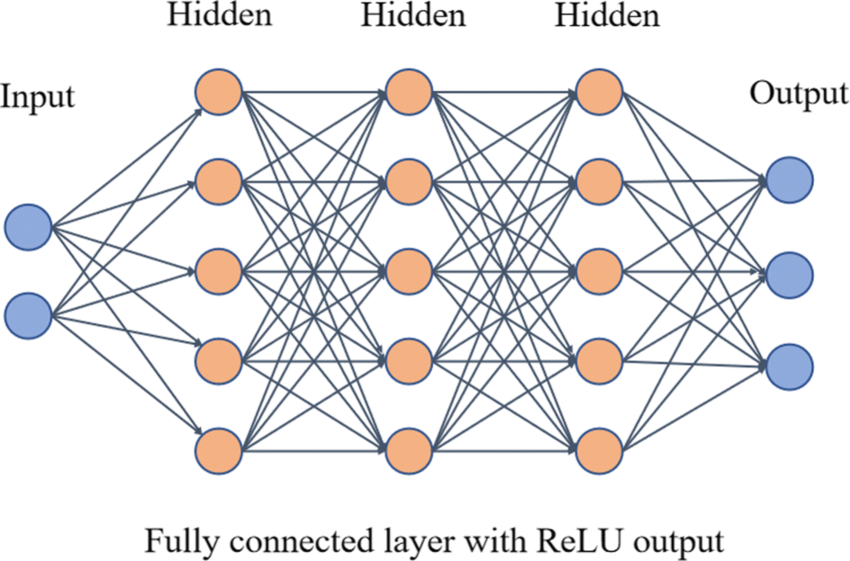
Các hàm tổng hợp phổ biến nhất có thể được áp dụng là:

* Max pooling: là một loại phép gộp được sử dụng trong mạng nơ-ron tích chập (CNN) và các kiến trúc liên quan. Nó thường được áp dụng sau lớp tích chập để giảm kích thước của đầu vào và tạo ra các phiên bản đã tạo ra của đặc trưng cục bộ.
* Sum pooling:  là một phương pháp trong xử lý ảnh và mạng học sâu, được sử dụng để giảm kích thước của feature maps trong quá trình trích xuất thông tin từ hình ảnh. Tương tự như Average Pooling và Max Pooling, Sum Pooling cũng là một loại phép gộp thông tin trong mạng neural.
* Average pooling : là một phép tích chập thường được sử dụng trong mạng học sâu và xử lý ảnh. Nó là một phần quan trọng của các kiến trúc mạng thần kinh như Convolutional Neural Networks (CNNs) để giảm kích thước của đặc trưng (feature maps) trong quá trình trích xuất thông tin từ hình ảnh.



##### c, Fully connected layers

Fully connected layer là lớp cuối cùng trong mạng CNN có chức năng thực hiện việc phân loại hoặc dự đoán các giá trị đầu ra của mô hình. Convolutional layer và pooling layer đã thực hiện các phép tích chập và giảm kích thước của dữ liệu đầu vào, tạo ra các đặc trưng cục bộ cho ảnh. Tuy nhiên thì chúng chỉ tạo ra các đặc trưng cục bộ và không liên kết các đặc trưng đó để đưa ra một dự đoán cuối cùng. Để phân loại chính xác hơn, chúng ta cần kết nối tất cả các đặc trưng đó với nhau và tính toán xác suất cho các lớp đầu ra. Fully connected layer trong mạng CNN được thiết kế có một số nơ-ron kết nối với tất cả các nơ-ron trong lớp trước đó (có thể là một lớp convolutional layer hoặc pooling layer) để nhận đầu vào là tất cả các đặc trưng từ lớp trước đó. Fully connected layer lúc này có nhiệm vụ kết nối các đặc trưng cục bộ đã được trích xuất từ các lớp convolutional layer và pooling layer để đưa ra một dự đoán cuối cùng.

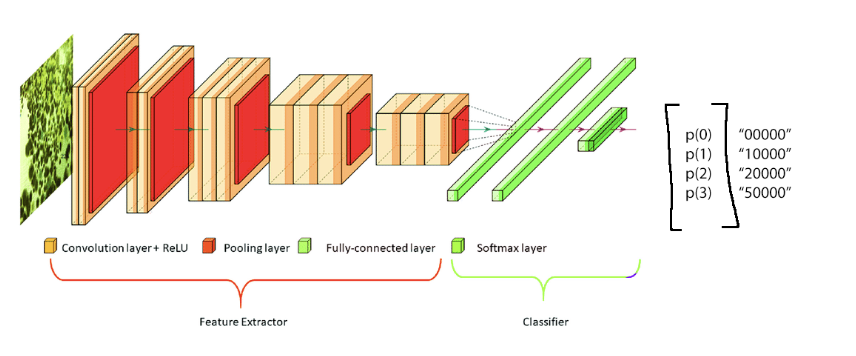


Hình 2.10: Minh họa lớp kết nối đầy đủ với đầu ra ReLU

Fully connected layer giúp mạng CNN học các mối quan hệ phức tạp giữa các đặc trưng đầu vào và tạo ra các phân loại chính xác hơn. Tuy nhiên khi sử dụng quá nhiều fully connected layer trong một mạng CNN sẽ dẫn đến hiện tượng overfitting, khi mô hình chỉ học nhớ dữ liệu huấn luyện mà không có khả năng dự đoán đúng trên các dữ liệu mới. Do đó số lượng fully connected layer cần được cân nhắc để tối ưu hóa hiệu suất của mạng CNN.

### Ví dụ minh họa mô hình CNN

Mô hình chung bài toán CNN: Input image -> Convolutional layer (Conv) + Pooling layer (Pool) -> Fully connected layer (FC) -> Output.



**Các lớp của mô hình được mô tả ,hoạt động theo thứ tự:**

* Ảnh đầu vào là ảnh màu, kích thước 28x28
* Lớp tích chập thứ nhất sử dụng 32 mặt nạ tích chập, padding=1, stride=1. Mặt nạ tích chập kích thước 3x3. Thu được ảnh đầu ra gồm 32 ảnh có kích thước 28x28 sau đó dùng hàm kích hoạt Rectified Linear Unit (ReLU)
* Lớp tích chập thứ nhất sử dụng 32 mặt nạ tích chập, padding=1, stride=1. Mặt nạ tích chập kích thước 3x3. Thu được ảnh đầu ra gồm 32 ảnh có kích thước 28x28 sau đó dùng hàm kích hoạt Rectified Linear Unit (ReLU)
* Lớp max pooling thứ nhất kích thước 2x2 cho đầu ra là 32 ảnh kích thước 14x14
* Tiếp theo dùng hàm FLATTEN để làm phẳng dữ liệu cho trước khi đến lớp Fully Connected , đặc trưng trong 32 ảnh tạo thành 1 vector có độ dài 14\*14\*32 = 6272 (nodes)
* Lớp full connect 1 (FC1) gồm 128 neural sau khi được phân lớp từ 6272 nodes dữ liệu đầu vào
* Lớp FC2 gồm 10 neural tương ứng đầu ra, các neural sẽ cho biết xác suất tương ứng với các lớp đối tượng của dữ liệu đầu vào

**Ưu điểm**

* Trích xuất đặc trưng tự động: CNN tự động học cách trích xuất các đặc trưng quan trọng từ dữ liệu ảnh mà không cần phải xác định rõ ràng các đặc trưng cụ thể. Điều này giúp giảm thiểu công việc thủ công và tốn thời gian để thiết lập các đặc trưng thủ công.
* Xử lý dữ liệu không gian: CNN được thiết kế đặc biệt để xử lý dữ liệu không gian như ảnh và âm thanh. Các lớp tích chập và gộp cho phép nó hiệu quả trong việc trích xuất thông tin từ các cấu trúc không gian này.
* Phân loại và nhận dạng tốt: CNN đã đạt được hiệu suất ấn tượng trong các nhiệm vụ phân loại và nhận dạng. Điều này bao gồm nhận biết khuôn mặt, phân loại đối tượng, dự đoán từ ảnh, và nhiều nhiệm vụ khác.
* Tích hợp thông tin cục bộ: Các lớp tích chập trong CNN cho phép mạng tập trung vào việc xử lý thông tin cục bộ trong các phần nhỏ của dữ liệu ảnh. Điều này giúp mạng nhận biết các đặc trưng cục bộ và tạo ra biểu diễn chất lượng cao.
* Khả năng học tập sâu: CNN có thể được thiết kế với nhiều lớp và kích thước khác nhau, cho phép nó học các đặc trưng ngày càng phức tạp và sâu hơn. Điều này giúp mạng học được các biểu diễn tổng hợp và trừu tượng từ dữ liệu ảnh.
* Hiệu suất cao: Các mô hình CNN tiên tiến có khả năng đạt được hiệu suất rất cao trong các thách thức như các cuộc thi nhận dạng hình ảnh (ImageNet) và các ứng dụng thực tiễn như xe tự hành.

**Nhược điểm**

* Số lượng tham số và tài nguyên tính toán: Các mạng CNN tiên tiến thường có hàng triệu hoặc thậm chí hàng tỷ tham số. Điều này đòi hỏi tài nguyên tính toán lớn để huấn luyện và triển khai, đặc biệt là trên các thiết bị có tài nguyên hạn chế như điện thoại di động.
* Overfitting: Trong mạng CNN lớn, có nguy cơ mô hình bị overfitting, tức là mô hình học quá nhiều từ dữ liệu huấn luyện và không tổng quát hóa tốt trên dữ liệu mới. Điều này có thể dẫn đến hiệu suất kém trên dữ liệu không nhìn thấy trước.
* Dữ liệu huấn luyện hạn chế: Mạng CNN yêu cầu một lượng lớn dữ liệu huấn luyện để học hiệu quả. Nếu dữ liệu không đủ đa dạng hoặc không đại diện, mô hình có thể không học được các biểu diễn tổng quát.
* Chuẩn bị dữ liệu: Trước khi áp dụng CNN, dữ liệu thường cần phải được chuẩn bị và tiền xử lý một cách cẩn thận, bao gồm việc thay đổi kích thước ảnh, chuẩn hóa dữ liệu và loại bỏ nhiễu.
* Khả năng biểu diễn không tốt cho dữ liệu không gian tương đối dài: Mạng CNN không thể hiệu quả biểu diễn thông tin về thứ tự và tương quan không gian dài hơn, chẳng hạn như các dãy thời gian dài.

# XÂY DỰNG HỆ QUYẾT ĐỊNH SỬ DỤNG MÔ HÌNH CNN TRONG BÀI TOÁN NHẬN DIỆN TIỀN VIỆT NAM

## Phát biểu bài toán

### Phát biểu bài toán

Tiền Việt Nam, với tư cách là một phương tiện biểu đạt tưởng tượng và tác động của con người, chứa đựng sức mạnh và độ phức tạp không ngừng. Trong môi trường số ngày nay, nhu cầu chuyển đổi dữ liệu từ hình ảnh tiền thành dạng văn bản có thể xử lý trở nên ngày càng quan trọng. Tính ứng dụng rộng rãi của việc nhận diện tiền trên ảnh mở ra những triển vọng mới cho nhiều lĩnh vực, bao gồm ngân hàng, thương mại, quản lý tài chính và an ninh.

Đối mặt với khối lượng lớn hình ảnh tiền từ nhiều nguồn khác nhau, việc tự động nhận diện tiền đưa ra một loạt thách thức. Tính đa dạng của các mệnh giá, hình ảnh, hoa văn, độ nghiêng, cỡ chữ, và nhiễu trong ảnh tạo nên môi trường phức tạp đòi hỏi sự đổi mới trong phương pháp nhận diện.

Ngoài ra, nhận diện tiền còn đối mặt với vấn đề của các dạng tiền giả, tiền cũ, và các dấu hiệu nhận biết đặc biệt có thể xuất hiện trên các tờ tiền. Điều này đặt ra yêu cầu cao đối với các hệ thống nhận diện, không chỉ phải xử lý đúng mệnh giá và đặc điểm của tiền mà còn phải duy trì độ chính xác khi có sự phức tạp và đa dạng trong ngữ cảnh.

Đồng thời, nhận diện tiền đóng vai trò quan trọng trong việc tạo ra dữ liệu có thể được sử dụng cho các ứng dụng thực tế, như kiểm tra tiền giả, tự động hóa các giao dịch ngân hàng, và nhiều ứng dụng khác. Vì vậy, việc nghiên cứu và xây dựng hệ thống nhận diện tiền chính xác và hiệu quả trở thành một thách thức quan trọng và hấp dẫn trong lĩnh vực trí tuệ nhân tạo và xử lý ảnh.

### Thuật toán áp dụng

Để thực hiện bài toán, nhóm chọn mô hình Convolutional Neural Network (CNN) cho bài toán nhận diện tiền Việt Nam vì nó có khả năng tự động học đặc trưng từ dữ liệu hình ảnh, giúp nắm bắt các đặc điểm phức tạp của tiền và các hoa văn trên tiền. Đặc biệt, lớp convolutional của CNN giúp nó hiệu quả trong việc nhận diện các cấu trúc tiền và từ trong các hình ảnh tiền Việt Nam.

CNN không chỉ phù hợp với đa dạng của dữ liệu hình ảnh mà còn có khả năng giảm chiều dữ liệu thông qua lớp pooling, giúp tối ưu hóa hiệu suất mô hình và giảm lượng thông tin không quan trọng. Sự tích hợp dễ dàng của các lớp cấp cao như LSTM trong kiến trúc CNN làm cho nó linh hoạt và mạnh mẽ trong việc xử lý các chuỗi kí tự và nhận diện cấu trúc phức tạp trong văn bản.

Khả năng tích hợp này cùng với hiệu suất tính toán cao của CNN làm cho nó trở thành một lựa chọn xuất sắc cho nhận diện tiền Việt Nam, đặc biệt là khi triển khai trong các ứng dụng thực tế đòi hỏi hiệu suất và độ chính xác cao. Tóm lại, CNN không chỉ cung cấp khả năng tự động hóa mà còn giúp tối ưu hóa quá trình nhận diện tiền Việt Nam, làm nổi bật sự hiệu quả và độ linh hoạt của nó trong ứng dụng thực tế.

## Cơ sở dữ liệu

Bộ dữ liệu được nhóm tìm kiếm trên trang Kaggle.com – một trang uy tín và có lượng dữ liệu phong phú phục vụ cho việc nghiên cứu và thử nghiệm các mô hình học sâu. Tên bộ dữ liệu:

Dataset from Vietnamese Currencies Classification 96.76

Nguồn gốc và Mục tiêu:

* Bộ dữ liệu này chứa toàn bộ tư liệu đào tạo của Kaggle.com cho việc nhận diện hình ảnh nhận diện mệnh giá tiền Việt Nam. Mục tiêu chính là hỗ trợ nghiên cứu về xử lý hình ảnh và OCR (Optical Character Recognition).

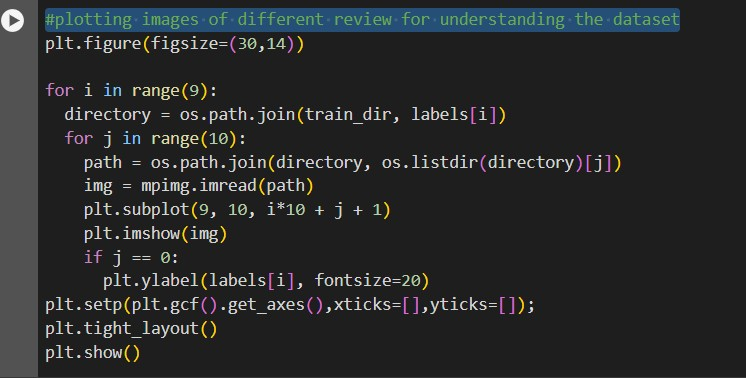
Nội dung của bộ dữ liệu:

* Chứa 9 mệnh giá tiền khác nhau là: 1.000VND, 2.000VND, 5.000VND, 10.000VND, 20.000VND, 50.000VND, 100.000VND, 200.000VND, 500.000VND.
* Bao gồm hơn 2,241 hình ảnh tiền được cắt ra từ các mệnh giá tiền và các góc chụp khác nhau.
* Cung cấp thông tin phân loại đúng (ground truth) cho các hình ảnh để sử dụng trong quá trình huấn luyện mô hình.

Đặc điểm của bộ dữ liệu:

* Bao gồm 9 mệnh giá tiền khác nhau mỗi mệnh giá có 259 ảnh chụp khác nhau, giúp đa dạng hóa dữ liệu.
* Hơn 2,241 hình ảnh tiền được phân loại một cách chính xác thông qua kiểm tra bằng tay.
* Bộ dữ liệu đã được làm sạch và chuẩn hoá sẵn giúp cho việc trainning trở nên dễ dàng hơn.

Để xem được dữ liệu dưới dạng ảnh, ta thực hiện lên sau:

Kết quả:



## Cài đặt

### Tiền xử lý dữ liệu

**Import các thư viện Python cần dùng:**

import os

import numpy as np

import matplotlib.pyplot as plt

import matplotlib.image as mpimg

import tensorflow as tf

from tensorflow import keras

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from keras.models import Sequential

from keras.layers import Dense, Conv2D, MaxPooling2D, Flatten

* NumPy (import numpy as np):

NumPy là một thư viện Python chuyên về xử lý mảng và ma trận.

np là một bí danh phổ biến được sử dụng để tham chiếu đến NumPy trong mã nguồn.

Các chức năng của NumPy bao gồm các phép toán số học, thống kê, và cung cấp đối tượng mảng nhanh và hiệu quả.

* Matplotlib (import matplotlib.pyplot as plt):

Matplotlib là một thư viện vẽ đồ thị và biểu đồ.

plt là bí danh phổ biến cho module pyplot trong Matplotlib, giúp thuận tiện cho việc vẽ đồ thị.

* TensorFlow (from tensorflow.keras.preprocessing.image import ImageDataGenerator):

TensorFlow là một thư viện mã nguồn mở của Google dành cho machine learning và deep learning.

ImageDataGenerator là một công cụ trong TensorFlow để tạo dữ liệu ảnh mới từ dữ liệu ảnh hiện có, thường được sử dụng trong quá trình augmentation.

* TensorFlow Keras:

Keras là một high-level neural networks API cho TensorFlow.

Sequential là một kiểu mô hình neural network cho phép xây dựng mô hình theo kiểu tuần tự, lớp theo lớp.

Dense, Dropout, Flatten, Conv2D, MaxPooling2D là các lớp cơ bản được sử dụng để xây dựng kiến trúc mô hình.

RMSprop là một trình tối ưu hóa được sử dụng để cập nhật trọng số của mô hình dựa trên giảm độ lớn của gradient.

Đọc dữ liệu từ file và gắn nhãn

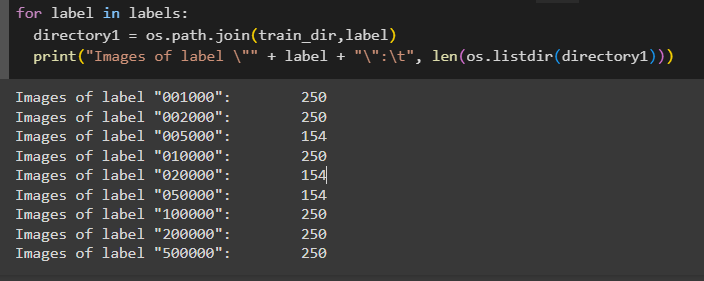
train\_dir = os.path.join('/content/drive/MyDrive/hhtqd/dataset/train')

labels = ['001000', '002000', '005000', '010000', '020000','050000','100000','200000','500000']

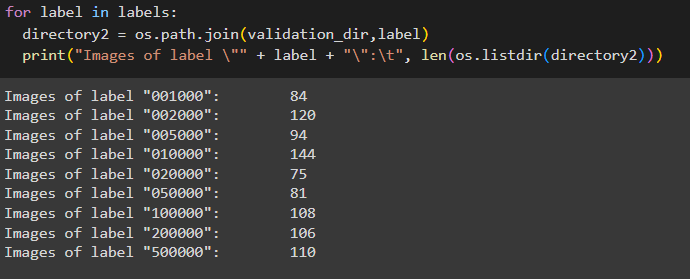
validation\_dir = os.path.join('/content/drive/MyDrive/hhtqd/dataset/validation')

labels = ['001000', '002000', '005000', '010000', '020000','050000','100000','200000','500000']

Kiểm tra size của dữ liệu train:



Kiểm tra size của dữ liệu validation:



- Tiền xử lý ảnh với ImageDataGenerator:

from tensorflow.keras.preprocessing.image import ImageDataGenerator

train\_data\_gen = ImageDataGenerator(rescale=1/255,)

validation\_data\_gen = ImageDataGenerator(rescale=1/255)

ImageDataGenerator là lớp học của Keras để tổng hợp dữ liệu hình ảnh cho học sâu. Nó cho phép dễ dàng truy cập vào hệ thống tệp cục bộ của bạn và nhiều phương pháp khác nhau để tải dữ liệu từ các cấu trúc khác nhau. Nó cũng có một số khả năng xử lý trước và tăng cường dữ liệu khá mạnh mẽ. Hàm này dùng để chuyển từ file ảnh 35 JPG thành dữ liệu tensor cho keras, cho phép sinh mẫu ngẫu nhiên bằng cách biến dạng, gây nhiễu ngẫu nhiên ảnh gốc. Tùy chỉnh tham số rescale cho phép chuyển giá trị pixel từ 0:255 thành thang đo [0:1] để thích hợp cho mạng neuron.

Tham số rescale = 1./255 có tác dụng chuẩn hóa dữ liệu ảnh, đưa toàn bộ giá trị điểm ảnh về các giá trị nằm trong khoảng [0, 1]. Việc chuẩn hóa dữ liệu này giúp quá trình huấn luyện mô hình được nhanh hơn.

train\_generator = train\_data\_gen.flow\_from\_directory(train\_dir,                                                   target\_size=(144,144),

                                                    batch\_size=32,

                                                 class\_mode='binary')

validation\_generator = validation\_data\_gen.flow\_from\_directory(validation\_dir,                                                target\_size=(144,144),

                                           batch\_size=32,

                                     class\_mode='binary')

Target size là kích thước của các ảnh được đưa vào tập huấn luyện Train, biến

đổi chúng về cùng một kích thước (144, 144).

Class mode là tính chất của nhãn kết quả = categorical (bài toán phân loại)

### Tiến hành huấn luyện

from keras.models import Sequential

from keras.layers import Dense, Conv2D, MaxPooling2D, Flatten

model = tf.keras.models.Sequential()

#lớp CNN1

model.add(Conv2D(32,(3,3), activation='relu', input\_shape=(144,144,3)))

model.add(MaxPooling2D(2,2))

#lớp CNN2

model.add(Conv2D(64,(3,3), activation='relu'))

model.add(MaxPooling2D(2,2))

#lớp CNN3

model.add(Conv2D(128,(3,3), activation='relu'))

model.add(MaxPooling2D(2,2))

#chuyển dữ liệu đầu ra của mạng CNN từ mảng 2 chiều về mảng 1 chiều

model.add(Flatten())

#lớp ẩn

model.add(Dense(512,activation=tf.nn.relu))

#lớp output

model.add(Dense(9,activation=tf.nn.softmax))

model.summary()

Trong đó:

* Sequential : cho phép người dùng tạo ra một mô hình học sâu bằng cách xếp chồng các lớp (layer) đơn giản lên nhau, trong đó mỗi lớp chỉ có một tensor đầu vào và một tensor đầu ra.
* Dense: là một lớp kết nối đầy đủ (fully connected layer), tức là mỗi nơ-ron (unit) của lớp này được nối với tất cả các nơ-ron của lớp trước và lớp sau.
* Conv2D:  Conv2D là một lớp tích chập 2 chiều (2D convolution layer). Tham số đầu tiên (32) là số bộ lọc và tham số thứ hai (3,3) là kích thước bộ lọc.
* MaxPooling2D: là một lớp (layer) dùng để giảm kích thước đầu vào bằng cách lấy giá trị lớn nhất trong một cửa sổ nhỏ (pooling window) trên mỗi kênh của đầu vào.
* Flatten: chuyển ảnh từ dạng ma trận về mảng 1 chiều.

Mô hình gồm 6 tầng: Input image -> CNN1 -> CNN2 -> CNN3-> Fully connected layer -> Output.

Mô hình có 3 lớp CNN:

- Lớp CNN1 gồm 32 bộ lọc kích thước 3x3, kết nối với đầu vào nên mô tả rõ thông tin của đầu vào (input\_shape).

- Lớp CNN2 gồm 64 bộ lọc kích thước 3x3

- Lớp CNN3 gồm 128 bộ lọc kích thước 3x3

Với mỗi bộ lọc khác nhau sẽ học được những đặc trưng khác nhau của ảnh, do đó mỗi tầng convolutional ta sẽ dùng nhiều bộ lọc (CNN1 sử dụng 32 bộ lọc, CNN2 sử dụng 64 bộ lọc, CNN3 sử dụng 128 bộ lọc) để học được nhiều đặc trưng của ảnh (ví dụ biên ngang, biên dọc…). Mỗi lớp CNN làm nhiệm vụ lấy đặc trưng của ảnh và nén ảnh từ kích cỡ ảnh to về ảnh nhỏ. Từ đó để giảm bớt số lượng nơron cần phải lưu trữ, giảm bớt độ phức tạp của thuật toán.

* Hidden layer (Lớp ẩn): chứa 512 neuron. Một lớp ẩn nằm giữa đầu vào và đầu ra của thuật toán. Các lớp ẩn thực hiện các phép biến đổi phi tuyến của các đầu vào được đưa vào mạng. Các lớp ẩn cho phép chức năng của mạng nơ-ron được chia nhỏ thành các phép biến đổi dữ liệu cụ thể. Mỗi chức năng lớp ẩn được chuyên biệt hóa để tạo ra một đầu ra xác định. Ví dụ: các chức năng của lớp ẩn được sử dụng để xác định hình dạng rác thải có thể được sử dụng kết hợp với các lớp tiếp theo để nhận dạng hình ảnh rác thải.
* Output layer (Lớp đầu ra): sử dụng hàm softmax tính toán xác suất xảy ra của một sự kiện. Hàm softmax sẽ tính khả năng xuất hiện của một class trong tổng số tất cả các class có thể xuất hiện. Sau đó, xác suất này sẽ được sử dụng để xác định class mục tiêu cho các input. Cụ thể, hàm softmax biến vector k chiều có các giá trị thực bất kỳ thành vector k chiều có giá trị thực có tổng bằng 1. Giá trị nhập có thể dương, âm, bằng 0 hoặc lớn hơn 1, nhưng hàm softmax sẽ luôn biến chúng thành một giá trị nằm trong khoảng (0:1].

model.compile(loss = 'sparse\_categorical\_crossentropy',

              optimizer=keras.optimizers.Adam(learning\_rate =0.0001),

              metrics=['accuracy'])

* Compile: chọn các tham số để huấn luyện mô hình.
* Optimizer: thuật toán tối ưu là cơ sở để xây dựng mô hình neural network với mục đích “học ” được các features ( hay pattern) của dữ liệu đầu vào, từ đó có thể tìm 1 cặp weights và bias phù hợp để tối ưu hóa model.
* Adam: Nhóm sử dụng Optimizer Adam cho ra độ chính xác cao nhất trong nhiều lần thử nghiệm so với RMSprop. Adam là một phương pháp tối ưu hóa dựa trên gradient ngẫu nhiên (stochastic gradient descent) mà dựa trên ước lượng thích ứng của các mô-men bậc nhất và bậc hai. Adam có thể được coi là sự kết hợp của RMSprop và SGD với động lượng. Adam là một trong những thuật toán tối ưu hóa phổ biến nhất cho các mô hình học sâu (deep learning) trong Keras. Trong đó có sử dụng tham số learning\_rate: tốc độ học, là một số thực dương, quyết định bước nhảy của thuật toán. Mặc định là 0.001.
* Loss: hàm tính toán sai số giữa giá trị học được và giá trị thực tế, sử dụng sparse\_categorical\_crossentropy trong trường hợp dự đoán nhiều lớp.
* Metrics: thước đo để ta đánh giá accuracy mô hình

- Huấn luyện mô hình

class myCallback(tf.keras.callbacks.Callback):

  def on\_epoch\_end(self, epoch, logs={}):

    if(logs.get('accuracy')==1):

      print("\nReached  100% accuracy so cancelling training!")

      self.model.stop\_training = True

callbacks = myCallback()

history = model.fit(train\_generator,

                    epochs = 50, verbose = 1,

                    validation\_data = validation\_generator,

                    callbacks=[callbacks])

Đối với mô hình này, nhóm chúng tôi đã thử nghiệm với nhiều tham số batch\_size, steps\_per\_epoch, epochs, validation\_steps và đã chọn ra tham số phù hợp để có được kết quả tốt nhất, có thể dự đoán chính xác tên các loài động vật. epochs = 20 là số vòng lặp của mô hình.

History=model.fit: Mỗi khi dùng model.fit (), nó sẽ trả về một đối tượng hard.callbacks. History giúp ghi lại các biến.

Verbose là tham số quyết định mức độ thông tin mà bạn muốn nhận được trong quá trình huấn luyện. Tham số verbose có thể có giá trị là 0, 1 hoặc 2:

* verbose = 0: không in ra bất kỳ thông tin nào.
* verbose = 1: in ra một thanh tiến trình (progress bar) và các thông tin về mất mát (loss) và độ chính xác (accuracy) của mô hình sau mỗi epoch (vòng lặp).
* verbose = 2: in ra các thông tin về mất mát và độ chính xác của mô hình sau mỗi epoch, nhưng không có thanh tiến trình.

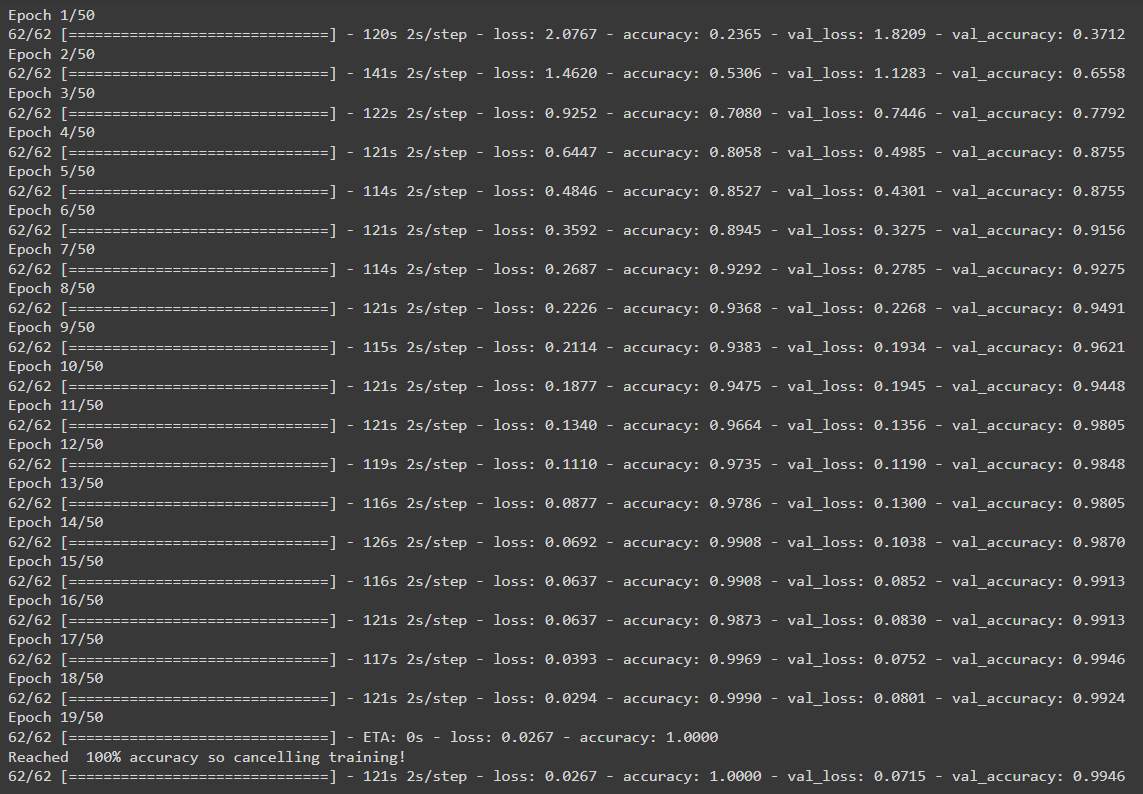
Verbose mặc định là 1 cho hầu hết các trường hợp.

Validation\_steps: Chỉ có liên quan nếu validation\_data được cung cấp và là tập dữ liệu. Tổng số bước (lô mẫu) cần vẽ trước khi dừng khi thực hiện xác nhận vào cuối mỗi Epoch.

Hàm callbacks được gọi khi mức độ chính xác accuracy==100% huấn luyện sẽ kết thúc.

- Xác định độ chính xác của mô hình

Sau khi chạy nhóm chúng tôi thu được kết quả như sau:



### Đánh giá và lưu mô hình

score = model.evaluate(validation\_generator, verbose=0)

print('Sai số kiểm tra là: ', score[0])

print('Độ chính xác kiểm tra là: ', score[1])

plt.figure(figsize=(10, 6))

plt.plot(history.history['loss'], label='Training Loss')

plt.plot(history.history['accuracy'], label='Training Accuracy')

plt.plot(history.history['val\_loss'], label='Validation Loss')

plt.plot(history.history['val\_accuracy'], label='Validation Accuracy')

plt.title('NhanDienTien\_Training')

plt.xlabel('Epoch')

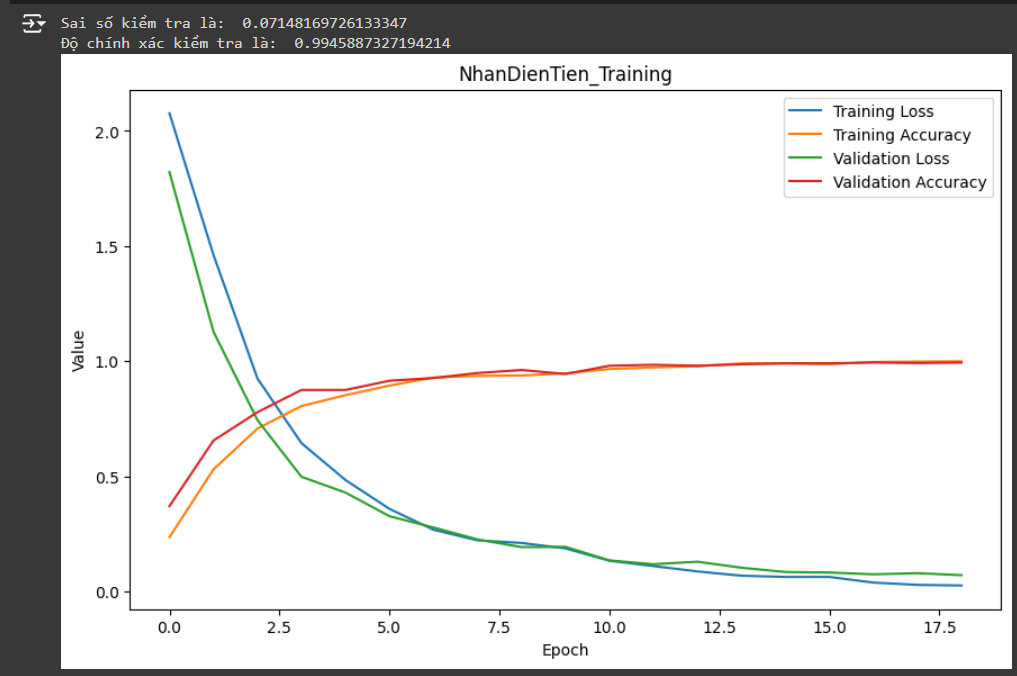
plt.ylabel('Value')

plt.legend()

plt.show()

Kết quả đánh giá mô hình:

* Sai số là xấp xỉ 0.0715
* Độ chính xác là xấp xỉ 0.9946

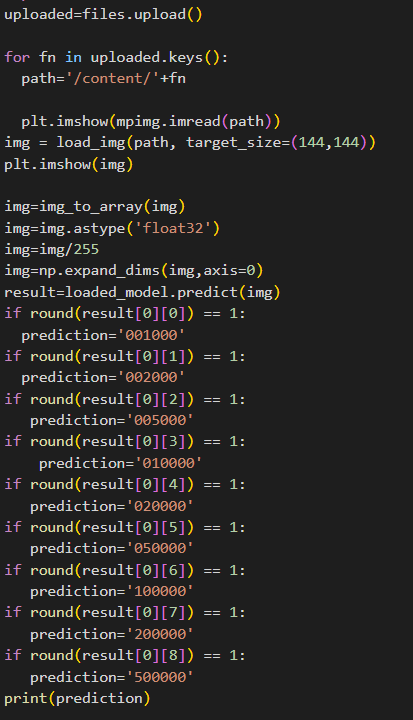


* Lưu model:

Lưu model dưới dạng file .h5 để lưu kết quả đã train ở trên để sử dụng cho test mô hình.

model.save('/content/drive/MyDrive/hhtqd/nhandientien.h5')

### Test mô hình



- Đầu tiên cần import file từ google colab

- From keras.preprocessing import image : Sau khi phân lớp, đầu ra của net là số thứ tự của lớp (1,2,3,…) nên ta cần decode kết quả tùy thuộc vào bộ dữ liệu sử dụng và thư viện tiền xử lý đầu vào của file.

- Matplotlib: để thực hiện các suy luận thống kê cần thiết, cần phải trực quan hóa dữ liệu của bạn và Matplotlib là một trong những giải pháp như vậy cho người dùng Python.

- Uploaded=files.upload() : Tải file từ tệp test

- Path='/content/'+fn : Load ảnh lên và xử lý trước khi đưa vào để phân lớp

- Plt.imshow(mpimg.imread(path)) : In ảnh ra để đọc

- Img=image.load\_img(path,target\_size=(144,144)) : đặt kích thước của ảnh

- img=img\_to\_array(img), img=img.astype('float32'), img=img/255, img=np.expand\_dims(img,axis=0) : Tiền xử lý ảnh

- Sử dụng loaded\_model.predict(img) để nhận diện ảnh bằng mô hình đã được train trước đó.

- Print(prediction]): In giá trị dự đoán.

Chạy mô hình và chọn một bức ảnh tiền bất kì trên tập dữ liệu Test hoặc đã tải về trên máy, kết quả nhận diện hình ảnh tiền sẽ dựa trên những gì mô hình được huấn luyện, như sau:



Kết quả nhận diện là giá trị tờ tiền.

**Kết luận:** Chương ba đã trình bày quá trình xây dựng hệ quyết định cho bài toán nhận diện chữ viết sử dụng mô hình Convolutional Neural Network (CNN). Qua nghiên cứu và thực hiện, chúng tôi đã đạt được những kết quả đáng chú ý về hiệu suất của hệ thống. Mô hình CNN đã được chọn lựa và tinh chỉnh để đáp ứng đặc điểm riêng của bài toán nhận diện chữ viết. Việc này bao gồm việc xử lý đối với độ phức tạp của các biểu diễn hình ảnh, đồng thời tối ưu hóa khả năng trích xuất đặc trưng từ vùng quan trọng của hình ảnh chữ viết. Tóm lại, nội dung chương đã đặt nền tảng cho việc triển khai một hệ thống nhận diện tiền hiệu quả dựa trên mô hình CNN. Các kết quả đạt được không chỉ là bước quan trọng trong nghiên cứu của chúng tôi mà còn mang lại những triển vọng tích cực cho ứng dụng thực tế trong lĩnh vực xử lý và nhận diện tiền.

# TỔNG KẾT

Kết thúc hành trình nghiên cứu và thực hiện đề tài "Xây dựng hệ quyết định sử dụng KNN ứng dụng trong gợi ý theo thể loại phim", chúng ta đã có những cái nhìn sâu rộng về tiềm năng và ứng dụng của mô hình K-Nearest Neighbors (KNN) trong lĩnh vực gợi ý phim.

Qua quá trình tìm hiểu và áp dụng, chúng ta đã thấy rằng KNN, mặc dù đơn giản về cơ bản, nhưng lại mang lại hiệu quả cao khi xử lý dữ liệu lớn và đề xuất gợi ý dựa trên những thông tin gần gũi và phù hợp nhất với người dùng. Khả năng phân loại và đề xuất dựa trên sự tương đồng của các đối tượng trong không gian đặc trưng đã giúp hệ thống của chúng ta đưa ra những gợi ý phim chính xác và thú vị.

Tuy nhiên, như mọi mô hình học máy khác, KNN cũng không tránh khỏi những hạn chế. Việc tiền xử lý dữ liệu, chọn lựa đặc trưng phù hợp và tinh chỉnh tham số sẽ ảnh hưởng đến độ chính xác và hiệu suất của mô hình. Điều này đặt ra yêu cầu về việc nghiên cứu và phát triển liên tục, để đảm bảo hệ thống gợi ý luôn cập nhật và phản ánh đúng nhu cầu của người dùng.

Trong tương lai, để cải thiện và mở rộng ứng dụng, chúng ta có thể xem xét việc kết hợp KNN với các mô hình học máy khác, hoặc áp dụng các kỹ thuật tiên tiến hơn trong lĩnh vực học máy để tối ưu hóa hiệu suất và đáp ứng nhu cầu ngày càng đa dạng của thị trường phim.

Tóm lại, đề tài này không chỉ mở ra cơ hội để khám phá và áp dụng mô hình KNN trong gợi ý phim, mà còn thúc đẩy chúng ta suy ngẫm về tiềm năng của trí tuệ nhân tạo và học máy trong việc cải thiện trải nghiệm người dùng. Cảm ơn tất cả mọi người đã đồng hành và đóng góp ý kiến trong quá trình này. Chúng ta hãy tiếp tục khám phá và phát triển công nghệ, mang lại giá trị thực sự cho cộng đồng.

# TÀI LIỆU THAM KHẢO

1. Trường Đại học Công Nghiệp Hà Nội, *Giáo trình hệ hỗ trợ quyết định*.
2. JavatPoint, *K-Nearest Neighbor(KNN) Algorithm for Machine Learning.* [www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning](http://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning).
3. Geeksforgeeks, K-Nearest Neighbor(KNN) Algorithm, [www.geeksforgeeks.org/k-nearest-neighbours/](http://www.geeksforgeeks.org/k-nearest-neighbours/).
4. Python, *Python interface to Tcl/Tk*, [docs.python.org/3/library/tkinter.html](https://docs.python.org/3/library/tkinter.html).