**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN KHAI THÁC DỮ LIỆU**

**ĐỀ TÀI:**

**ĐỀ TÀI: PHÂN LOẠI NGUỒN GỐC CỦA ẢNH**

**GVHD: ThS. Hà Lê Hoài Trung**

**Lớp: IS252.N21**

Nhóm sinh viên thực hiện:

Võ Phạm Thùy Nhung MSSV: 20520679

Lương Phúc Hậu MSSV: 20521298

Nguyễn Hữu Thắng MSSV: 20520759

**TP. HỒ CHÍ MINH, NĂM 2023**

**LỜI CẢM ƠN**

Kính gửi Thầy Hà Lê Hoài Trung,

Chúng em xin gửi đến Thầy báo cáo cuối kỳ môn Khai Thác Dữ Liệu. Trong suốt quá trình học tập và nghiên cứu trong môn học này, chúng em đã học được rất nhiều kiến thức và kỹ năng quan trọng liên quan đến việc khai thác dữ liệu, từ việc thu thập dữ liệu đến phân tích và trích xuất thông tin.

Trước khi tiến hành nghiên cứu và thực hiện báo cáo này, chúng em đã nắm vững các khái niệm cơ bản về khai thác dữ liệu, bao gồm các phương pháp, kỹ thuật và công cụ sử dụng để xử lý dữ liệu lớn và tìm ra những thông tin quan trọng. Chúng em cũng đã áp dụng những kiến thức đã học vào việc thực hiện các bài toán và dự án thực tế.

Trong báo cáo này, chúng em sẽ trình bày kết quả của quá trình nghiên cứu và áp dụng khai thác dữ liệu vào một bài toán cụ thể. Chúng em đã tiến hành thu thập dữ liệu, tiền xử lý dữ liệu, áp dụng các phương pháp khai thác dữ liệu phù hợp và trình bày kết quả qua các biểu đồ, bảng số liệu và phân tích.

Chúng em xin chân thành cảm ơn Thầy vì sự hướng dẫn và hỗ trợ trong suốt quá trình học tập. Thầy đã chia sẻ kiến thức rộng lớn và cung cấp những phản hồi hữu ích, giúp chúng em nắm bắt được bản chất và ý nghĩa của môn học.

Chúng em hy vọng rằng báo cáo này sẽ thể hiện được sự hiểu biết và ứng dụng thực tế của chúng em với kiến thức đã học. Chúng em sẽ rất mong nhận được sự đánh giá và những góp ý từ Thầy để chúng em có thể tiếp tục nâng cao và phát triển kỹ năng khai thác dữ liệu của mình trong tương lai.

Xin chân thành cảm ơn Thầy và mong nhận được sự chỉ dẫn của Thầy về báo cáo này.

*Thành phố Hồ Chí Minh, tháng 05 năm 2023*

***Nhóm sinh viên thực hiện***

**BẢNG PHÂN CÔNG, ĐÁNH GIÁ THÀNH VIÊN:**

Bảng phân công, đánh giá thành viên

|  |  |  |  |
| --- | --- | --- | --- |
|  | Lương Phúc Hậu | Võ Phạm Thùy Nhung | Nguyễn Hữu Thắng |
| Chọn dữ liệu, thảo luận và đánh giá dữ liệu | x | x | x |
| Mô tả dữ liệu, mô tả bài toán | x | x | x |
| Tiền xử lý dữ liệu | x |  |  |
| Thực hiện giải thuật SVM |  | x |  |
| Thực hiện giải thuật Deep Learning |  |  | x |
| Đánh giá thời gian, độ chính xác của thuật toán | x |  |  |
| Thực hiện dự báo kết quả bằng thuật toán tối ưu nhất |  | x |  |
| Xây dựng tài liệu, viết báo cáo, tìm tài liệu tham khảo | x | x | x |

**MỤC LỤC**

[DANH MỤC HÌNH ẢNH 3](#_Toc135906515)

[DANH MỤC TỪ VIẾT TẮT 4](#_Toc135906516)

[CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI 5](#_Toc135906517)

[1.1. Lý do chọn đề tài 5](#_Toc135906518)

[1.2. Phát biểu bài toán 6](#_Toc135906519)

[CHƯƠNG 2. TỔNG QUAN VỀ DỮ LIỆU 7](#_Toc135906520)

[2.1. Giới thiệu Dataset 7](#_Toc135906521)

[2.2. Tiền xử lý dữ liệu 8](#_Toc135906522)

[CHƯƠNG 3. PHƯƠNG PHÁP TIẾP CẬN 10](#_Toc135906523)

[3.1. Support Vector Machine (SVM) 10](#_Toc135906524)

[3.1.1 Định nghĩa 10](#_Toc135906525)

[3.1.2 Cách thức hoạt động 10](#_Toc135906526)

[3.1.3 SVM đa lớp (Multiclass SVM) 11](#_Toc135906527)

[3.2. Convolutional Neural Network (CNN) 14](#_Toc135906528)

[3.2.1 Định nghĩa 14](#_Toc135906529)

[3.2.2 Cấu trúc 16](#_Toc135906530)

[CHƯƠNG 4. THỰC NGHIỆM 18](#_Toc135906531)

[4.1. Quá trình huấn luyện 18](#_Toc135906532)

[4.1.1 Mô hình SVM 18](#_Toc135906533)

[4.1.2 Mô hình CNN 19](#_Toc135906534)

[4.2. Phân tích và so sánh 21](#_Toc135906535)

[4.2.1 Mô hình SVM 21](#_Toc135906536)

[4.2.2 Mô hình CNN 22](#_Toc135906537)

[4.2.3 So sánh 2 mô hình 25](#_Toc135906538)

[CHƯƠNG 5. TỔNG KẾT 27](#_Toc135906539)

[5.1. Kết quả đạt được 27](#_Toc135906540)

[5.1.1 Mô hình SVM 27](#_Toc135906541)

[5.1.2 Mô hình CNN 27](#_Toc135906542)

[5.2. Hướng phát triển 28](#_Toc135906543)

[DANH MỤC TÀI LIỆU THAM KHẢO 29](#_Toc135906544)

# DANH MỤC HÌNH ẢNH

[Hình 1. Hình minh họa SVM 10](#_Toc135905755)

[Hình 2. Ví dụ phân loại SVM 12](#_Toc135905756)

[Hình 3. Ví dụ cách tiếp cận One-to-One 13](#_Toc135905757)

[Hình 4. Ví dụ cách tiếp cận One-to-Rest 14](#_Toc135905758)

[Hình 5. Minh họa ma trận 3x3 15](#_Toc135905759)

[Hình 6. Minh họa ma trận 5x5 15](#_Toc135905760)

[Hình 7. Minh họa các lớp Filter 17](#_Toc135905761)

# DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Từ viết tắt** | **Nội dung** |
| **1** | **AI** | Artificial Intelligence |
| **2** | **SVM** | Support Vector Machine |
| **3** | **CNN** | Convolutional Neural Network |

1. GIỚI THIỆU ĐỀ TÀI
   1. Lý do chọn đề tài

Đồ họa máy tính (Computer Graphics) chắc hẳn đã không còn là khái niệm xa lạ. Đây là lĩnh vực nghiên cứu và ứng dụng việc tạo ra, xử lý và hiển thị hình ảnh trên máy tính. Đồ họa máy tính bao gồm nhiều phần khác nhau, từ việc tạo ra hình ảnh đơn giản như các hình học cơ bản đến việc tạo ra hình ảnh phức tạp như đồ họa 3D và hiệu ứng đặc biệt.

Tính đến hiện tại, công nghệ đồ họa máy tính đóng góp rất nhiều cho sự phát triển của các ngành công nghiệp và là một thành phần không thể thiếu trong quá trình sản xuất và thiết kế sản phẩm hiện đại. Chắc chắn rằng, với sự phát triển không ngừng nghỉ của công nghệ, đồ họa máy tính sẽ tiếp tục phát triển và trở thành một trong những công nghệ quan trọng nhất của thế giới hiện đại.

Trong thời gian gần đây, sự phát triển nhảy vọt của AI (Artificial Intelligence) đã mang lại một sự đổi mới và tiềm năng trên nhiều mặt của đời sống kinh tế, xã hội.

Trí tuệ nhân tạo (AI) nói một cách ngắn gọn là thành quả mô phỏng các quá trình trí tuệ của con người bằng máy móc, đặc biệt là các hệ thống máy tính. Trong lĩnh vực thiết kế nói riêng, công cụ thiết kế AI giúp hoàn thành thiết kế nhanh hơn bằng cách tự động tinh chỉnh chúng dựa trên hàng triệu thiết kế khác.

Với sự phát triển nhanh chóng và không ngừng của công nghệ AI và đồ họa máy tính, cùng những mặt tích cực mà chúng mang lại cũng có không ít tiêu cực. Đồ họa máy tính ngày càng chân thực và giống với ảnh chụp hơn, có thể được giả mạo dưới dạng ảnh chụp, gây ra các vấn đề nghiêm trọng về bảo mật. Sản xuất nghệ thuật bằng AI - mối đe dọa với bản quyền của họa sĩ từng là chủ đề được bàn tán sôi nổi ở nhiều quốc gia.

Nhận thấy được vấn đề nan giải đó, nhóm chúng em quyết định chọn đề tài phân biệt nguồn gốc của ảnh (tranh vẽ, ảnh kỹ thuật số, ảnh vẽ bởi AI, ảnh chụp).

* 1. Phát biểu bài toán

Với sự phát triển nhanh chóng của công nghệ, việc thiết kế cũng như tự động tạo ảnh càng trở nên tinh vi, nhìn bằng mắt thường chúng ta rất khó có thể phân biệt giữa tranh vẽ, ảnh kỹ thuật số, ảnh vẽ bởi AI và ảnh chụp.

Trong thực tế, việc phân biệt được nguồn gốc của một bức ảnh có thể giúp chúng ta đánh giá tính xác thực của ảnh và cung cấp thông tin quan trọng về quá trình tạo ra nó. Ví dụ, trong lĩnh vực nghệ thuật, nhận diện nguồn gốc tranh vẽ có thể đảm bảo tính xác thực và giá trị của tác phẩm. Trong lĩnh vực pháp y, nhận diện nguồn gốc của ảnh kỹ thuật số có thể hỗ trợ trong việc phát hiện gian lận hoặc chỉnh sửa ảnh không đúng quy trình. Đối với ảnh được tạo bởi trí tuệ nhân tạo, việc nhận diện có thể giúp chúng ta hiểu được khả năng sáng tạo và tự động của hệ thống AI. Cuối cùng, việc phân loại ảnh chụp từ nguồn ngoại vi giúp chúng ta biết được liệu ảnh có thể đáng tin cậy và không bị chỉnh sửa.

Bài toàn mà nhóm đặt ra sử dụng thuật toán Máy vector hỗ trợ - Support Vector Machine (SVM) và Mạng nơ-ron tích chập - Convolutional Neural Network (CNN) để tiến hành phân loại nguồn gốc của các hình ảnh trên.

1. TỔNG QUAN VỀ DỮ LIỆU
   1. Giới thiệu Dataset

Dataset được tự thu thập và bao gồm tổng cộng 4,668 bức ảnh, được chia thành 4 nhãn khác nhau để phục vụ cho bài toán nhận diện nguồn gốc ảnh. Mỗi nhãn đại diện cho một loại nguồn gốc ảnh cụ thể. Số lượng bức ảnh trong mỗi nhãn được phân bố như sau:

* + - 1. Nhãn 1: Tranh vẽ (1107 bức ảnh)
  + Đây là nhãn đại diện cho các bức ảnh là tranh vẽ thủ công được tạo ra bởi nghệ nhân hoặc họa sĩ.
  + Các bức ảnh trong nhãn này có thể bao gồm các loại tranh vẽ như tranh dầu, tranh sơn mài, tranh sơn dầu, tranh vẽ chì, màu nước, …
  + Nguồn thu thập: <https://www.kaggle.com/datasets/thedownhill/art-images-drawings-painting-sculpture-engraving>
    - 1. Nhãn 2: Ảnh kỹ thuật số (1100 bức ảnh)
  + Đây là nhãn đại diện cho các bức ảnh kỹ thuật được tạo ra từ các phần mềm đồ họa máy tính hoặc các thiết bị điện tử khác. Thay vì sử dụng bút, cọ và màu sơn truyền thống, người nghệ sĩ sử dụng bút cảm ứng, bảng vẽ điện tử hoặc máy tính bảng và phần mềm đồ họa để tạo ra các tác phẩm số hoá.
  + Các bức ảnh trong nhãn này có thể là ảnh chân dung, ảnh phong cảnh, các hình ảnh trừu tượng hoặc viễn tưởng.
  + Nguồn thu thập: <https://stock.adobe.com/vn/search?k=%22digital%20painting%22>
    - 1. Nhãn 3: Ảnh vẽ bởi AI (1282 bức ảnh)
  + Đây là nhãn đại diện cho các bức ảnh được tạo ra bởi trí tuệ nhân tạo (AI) hoặc các thuật toán máy học.
  + Các bức ảnh trong nhãn này có thể bao gồm các ảnh tạo ra bởi mô hình học sâu, mô hình sinh tự động, hoặc các thuật toán khác dựa trên trí tuệ nhân tạo.
  + Nguồn thu thập: <https://dalle2.gallery/#search>
    - 1. Nhãn 4: Ảnh chụp (1199 bức ảnh)
  + Đây là nhãn đại diện cho các bức ảnh được chụp từ nguồn ngoại vi như điện thoại di động, máy ảnh hoặc máy quay.
  + Các bức ảnh trong nhãn này có thể bao gồm các loại ảnh chụp phong cảnh, ảnh các vật thể, ảnh động vật hoặc ảnh chân dung
  + Nguồn thu thập: <https://www.kaggle.com/datasets/andrewmvd/animal-faces>

<https://www.kaggle.com/code/nulldata/deep-image-search-for-finding-similar-images>

<https://www.kaggle.com/datasets/l3llff/flowers>

Các bức ảnh trong dataset đã được thu thập và đánh nhãn bằng tay để đảm bảo tính chính xác. Đây là một tập dữ liệu đa dạng với mục đích phục vụ cho bài toán nhận diện nguồn gốc ảnh, và nó có thể được sử dụng để huấn luyện và đánh giá các mô hình máy học hoặc các thuật toán xử lý ảnh liên quan đến bài toán này.

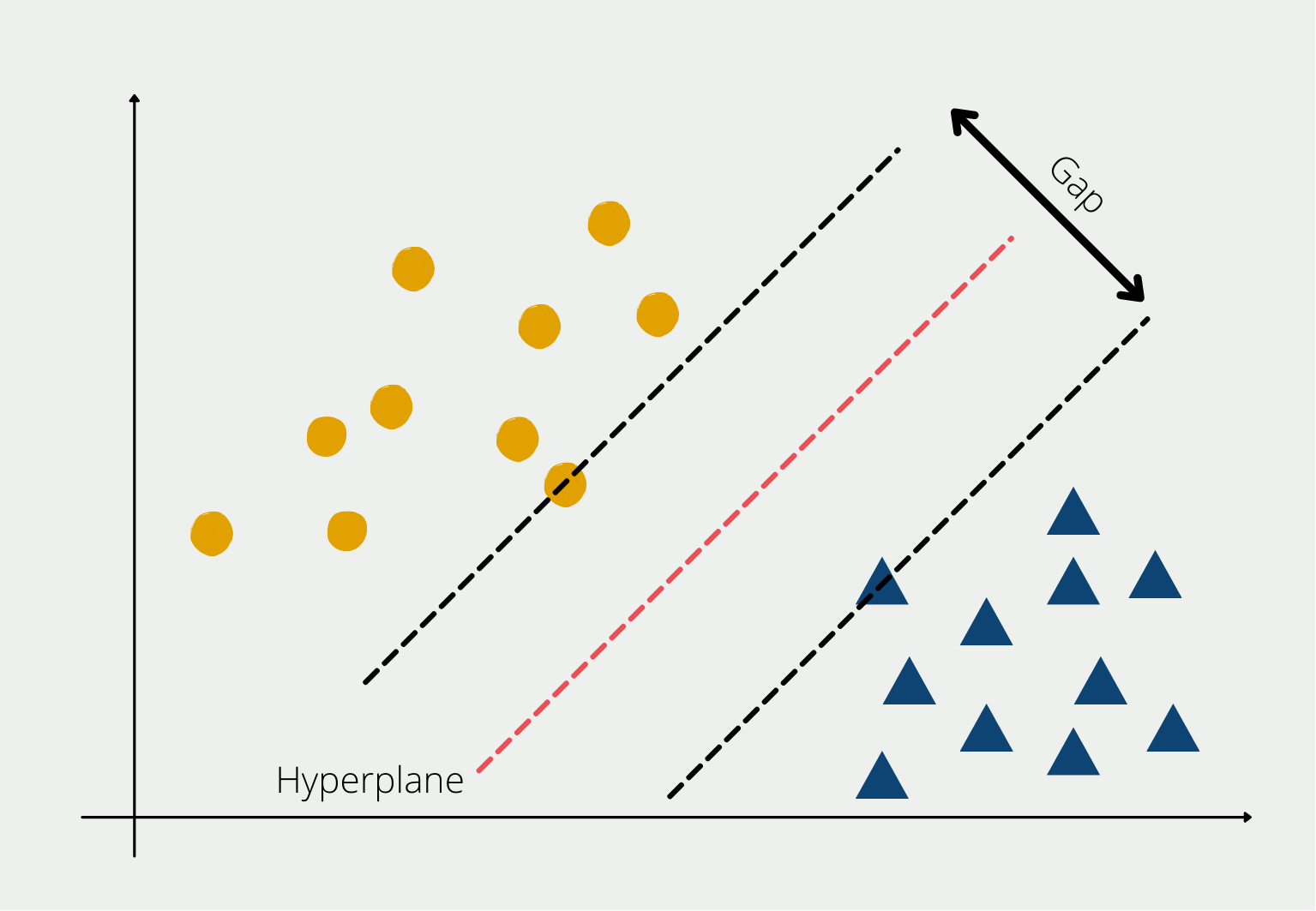
* 1. Tiền xử lý dữ liệu

Các bước tiền xử lý dữ liệu:

* Bước 1: Dữ liệu sau khi tải về sẽ chứa nhiều file rác trong quá khi crawl dữ liệu, cần tiến hành xóa bỏ các file này, chỉ giữ lại các file có định dạng là hình ảnh.
* Bước 2: Xóa bỏ những hình ảnh bị trùng lặp
* Bước 3: Chuẩn hóa hình ảnh về cùng kích thước. Ở đây nhóm chúng em chọn kích thước là (224,224), đây là 1 kích thước thường được sử dụng trong việc chuẩn hóa hình ảnh trong các bài toán máy học
* Bước 4: Chia các giá trị pixel của ảnh cho 255 để đưa chúng về khoảng từ 0 đến 1. Quá trình này còn được gọi là rescaling giúp chuẩn hóa dữ liệu ảnh
* Bước 5: Chia dataset thành 2 tập train và test theo tỉ lệ 8:2 để chuẩn bị cho việc áp dụng mô hình

1. PHƯƠNG PHÁP TIẾP CẬN
   1. Support Vector Machine (SVM)
      1. Định nghĩa

Máy vectơ hỗ trợ (Support Vector Machine - SVM) là một khái niệm trong thống kê và khoa học máy tính cho một tập hợp các phương pháp học có giám sát liên quan đến nhau để phân loại và phân tích hồi quy. SVM dạng chuẩn nhận dữ liệu vào và phân loại chúng vào hai lớp khác nhau. Do đó SVM là một thuật toán phân loại nhị phân. Với một bộ các ví dụ luyện tập thuộc hai thể loại cho trước, thuật toán luyện tập SVM xây dựng một mô hình SVM để phân loại các ví dụ khác vào hai thể loại đó. Một mô hình SVM là một cách biểu diễn các điểm trong không gian và lựa chọn ranh giới giữa hai thể loại sao cho khoảng cách từ các ví dụ luyện tập tới ranh giới là xa nhất có thể.



Hình 1. Hình minh họa SVM

* + 1. Cách thức hoạt động

Một máy vectơ hỗ trợ xây dựng một siêu phẳng hoặc một tập hợp các siêu phẳng trong một không gian nhiều chiều hoặc vô hạn chiều, có thể được sử dụng cho phân loại, hồi quy, hoặc các nhiệm vụ khác. Một cách trực giác, để phân loại tốt nhất thì các siêu phẳng nằm ở càng xa các điểm dữ liệu của tất cả các lớp (gọi là hàm Biên) càng tốt, vì nói chung Biên càng lớn thì sai số tổng quát hóa của thuật toán phân loại càng bé.

Trong nhiều trường hợp, không thể phân chia các lớp dữ liệu một cách tuyến tính trong một không gian ban đầu được dùng để mô tả một vấn đề. Vì vậy, nhiều khi cần phải ánh xạ các điểm dữ liệu trong không gian ban đầu vào một không gian mới nhiều chiều hơn, để việc phân tách chúng trở nên dễ dàng hơn trong không gian mới. Để việc tính toán được hiệu quả, ánh xạ sử dụng trong thuật toán SVM chỉ đòi hỏi tích vô hướng của các vectơ dữ liệu trong không gian mới có thể được tính dễ dàng từ các tọa độ trong không gian cũ. Tích vô hướng này được xác định bằng một hàm hạt nhân K(x,y) phù hợp. Một siêu phẳng trong không gian mới được định nghĩa là tập hợp các điểm có tích vô hướng với một vectơ cố định trong không gian đó là một hằng số. Vectơ xác định một siêu phẳng sử dụng trong SVM là một tổ hợp tuyến tính của các vectơ dữ liệu luyện tập trong không gian mới với các hệ số αi. Với siêu phẳng lựa chọn như trên, các điểm x trong không gian đặc trưng được ánh xạ vào một siêu mặt phẳng là các điểm thỏa mãn:

**Σiαi K(xi,x) = hằng số.**

Ghi chú rằng nếu K(x,y) nhận giá trị ngày càng nhỏ khi y xa dần khỏi x thì mỗi số hạng của tổng trên được dùng để đo độ tương tự giữa x với điểm xi tương ứng trong dữ liệu luyện tập. Như vậy, tác dụng của tổng trên chính là so sánh khoảng cách giữa điểm cần dự đoán với các điểm dữ liệu đã biết. Lưu ý là tập hợp các điểm x được ánh xạ vào một siêu phẳng có thể có độ phức tạp tùy ý trong không gian ban đầu, nên có thể phân tách các tập hợp thậm chí không lồi trong không gian ban đầu.

* + 1. SVM đa lớp (Multiclass SVM)

Ở dạng đơn giản nhất, SVM không hỗ trợ phân loại đa lớp nguyên bản. Nó hỗ trợ phân loại nhị phân và tách các điểm dữ liệu thành hai lớp. Đối với phân loại đa lớp, nguyên tắc tương tự được sử dụng sau khi chia lớp đa phân loại thành nhiều lớp phân loại nhị phân.

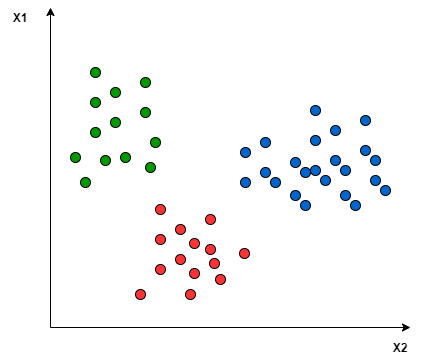
Ý tưởng là ánh xạ các điểm dữ liệu sang không gian nhiều chiều để đạt được sự phân tách tuyến tính lẫn nhau giữa hai lớp. Đây được gọi là cách tiếp cận One-to-One, chia nhỏ đa lớp thành nhiều các lớp phân loại nhị phân. Một bộ phân loại nhị phân cho mỗi cặp lớp.

Một cách tiếp cận khác mà người ta có thể sử dụng là One-to-Rest. Theo cách tiếp cận này, bảng phân tích được đặt thành một bộ phân loại nhị phân cho mỗi lớp.

Sự khác nhau giữa hai cách tiếp cận trên từ i tập dữ liệu lớp:

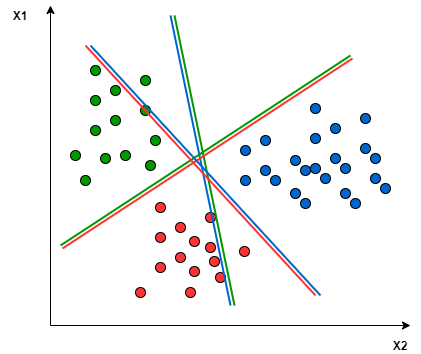
* Trong cách tiếp cận One-to-Rest, bộ phân loại có thể sử dụng ***m*** SVM. Mỗi SVM sẽ dự đoán tư cách thành viên của một trong ***m*** các lớp.
* Trong cách tiếp cận One-to-One, bộ phân loại có thể sử dụng SVM.

Ví dụ một bài toán phân loại 3 lớp; xanh lá cây, đỏ và xanh lam, như hình ảnh sau:



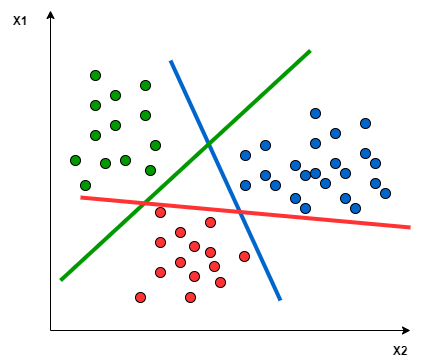
Hình 2. Ví dụ phân loại SVM

Với cách tiếp cận One-to-One, giữa hai lớp riêng biệt sẽ có một siêu mặt phảng để phân tách và lúc này sẽ bỏ qua điểm còn lại. Nhìn vào hình bên dưới, đường màu xanh dương đỏ phân tách hai lớp xanh dương và đỏ, bỏ qua các điểm thuộc lớp xanh lá. Điều này cũng tương tự với các bộ đôi còn lại.



Hình 3. Ví dụ cách tiếp cận One-to-One

Trong cách tiếp cận *One-to-Rest* , chúng ta cần một siêu phẳng để tách biệt giữa một lớp và tất cả các lớp khác cùng một lúc. Điều này có nghĩa là sự tách biệt có ảnh hưởng đến tất cả các điểm, chia chúng thành hai nhóm; một nhóm cho các điểm của lớp và một nhóm cho tất cả các điểm khác. Ví dụ: đường màu xanh lá cố gắng tối đa hóa khoảng cách giữa các điểm màu xanh lá và tất cả các điểm khác cùng một lúc:



Hình 4. Ví dụ cách tiếp cận One-to-Rest

* 1. Convolutional Neural Network (CNN)
     1. Định nghĩa

Mạng nơ-ron tích chập (Convolutional Neural Network - CNN) là một loại mạng nơ-ron nhân tạo được áp dụng phổ biến nhất để phân tích hình ảnh trực quan. CNN sử dụng một phép toán gọi là tích chập thay cho phép nhân ma trận chung trong ít nhất một trong các lớp của chúng. Chúng được thiết kế đặc biệt để xử lý dữ liệu pixel và được sử dụng trong quá trình nhận dạng và xử lý hình ảnh.

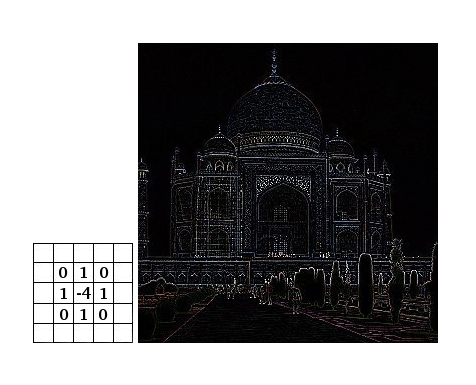
CNN được xem là một cửa sổ trượt (Sliding Windows) trên một ma trận như mô tả hình dưới:

Các convolutional layer có các parameter(kernel) đã được học để tự điều chỉnh lấy ra những thông tin chính xác nhất mà không cần chọn các feature.

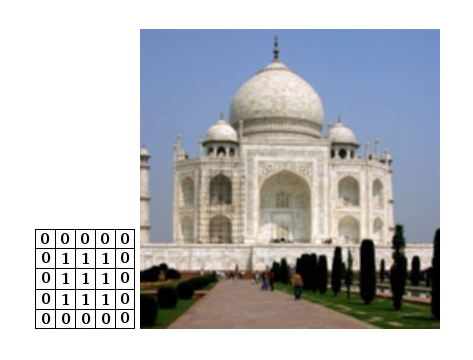
Trong hình ảnh ví dụ trên, ma trận bên trái là một hình ảnh trắng đen được số hóa. Ma trận có kích thước 5×5 và mỗi điểm ảnh có giá trị 1 hoặc 0 là giao điểm của dòng và cột.

Convolution hay tích chập là nhân từng phần tử trong ma trận 3. Sliding Window hay còn gọi là kernel, filter hoặc feature detect là một ma trận có kích thước nhỏ như trong ví dụ trên là 3×3.

Convolution hay tích chập là nhân từng phần tử bên trong ma trận 3×3 với ma trận bên trái. Kết quả được một ma trận gọi là Convoled feature được sinh ra từ việc nhận ma trận Filter với ma trận ảnh 5×5 bên trái.



Hình 5. Minh họa ma trận 3x3



Hình 6. Minh họa ma trận 5x5

* + 1. Cấu trúc

Một mạng thần kinh tích chập bao gồm một lớp đầu vào, các lớp ẩn và một lớp đầu ra. Trong mạng nơ ron tích chập, các lớp ẩn bao gồm một hoặc nhiều lớp thực hiện phép tích chập. Thông thường, điều này bao gồm một lớp thực hiện tích vô hướng của hạt tích chập với ma trận đầu vào của lớp.

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

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

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

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

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



Hình 7. Minh họa các lớp Filter

1. THỰC NGHIỆM
   1. Quá trình huấn luyện
      1. Mô hình SVM

A screen shot of a computer

Description automatically generated with medium confidence

Mô hình trên là một bộ phân loại (classifier) sử dụng thuật toán Support Vector Machine (SVM). Trong trường hợp này, mô hình sử dụng lớp SVC (Support Vector Classifier) trong thư viện scikit-learn.

Mô hình được tạo với các siêu tham số (hyperparameters) được chỉ định trong **parameters**. Cụ thể, có hai siêu tham số được sử dụng: 'gamma' và 'C'.

* 'gamma' là siêu tham số điều chỉnh hình dạng của siêu phẳng (hyperplane) tạo ra bởi SVM. Các giá trị được xác định là [0.01, 0.001, 0.0001].
* 'C' là siêu tham số đánh đổi sự phân loại chính xác và việc phân loại sai sót. Các giá trị được xác định là [1, 10, 100, 1000].

Để tìm ra các giá trị tốt nhất cho các siêu tham số này, mô hình sử dụng GridSearchCV (Grid Search Cross-Validation). Grid Search là một phương pháp tìm kiếm siêu tham số bằng cách duyệt qua tất cả các tổ hợp có thể của các giá trị siêu tham số được chỉ định và đánh giá mô hình trên từng tổ hợp này bằng cross-validation. GridSearchCV tự động thực hiện quá trình này trong một cách tối ưu.

Cuối cùng, sau khi hoàn thành quá trình Grid Search, **grid\_search** sẽ chứa thông tin về mô hình tốt nhất được tìm thấy, bao gồm các siêu tham số tốt nhất và hiệu suất tương ứng của mô hình trên dữ liệu đánh giá.

* + 1. Mô hình CNN

Mô hình CNN được xây dựng bằng cách sử dụng lớp Sequential từ thư viện tensorflow.keras.

A screen shot of a computer program

Description automatically generated with medium confidence

Mô hình có cấu trúc như sau:

* **layers.Conv2D(10,3,activation="relu", input\_shape=(224,224,3))**: Lớp tích chập với 10 bộ lọc (filters) có kích thước 3x3, hàm kích hoạt ReLU và kích thước đầu vào là (224, 224, 3). Đây là lớp đầu tiên trong mạng nơ-ron.
* **layers.Conv2D(10,3,activation="relu")**: Lớp tích chập thứ hai với 10 bộ lọc có kích thước 3x3, hàm kích hoạt ReLU. Lớp này không cần chỉ định kích thước đầu vào vì nó sẽ tự động nhận đầu vào từ lớp trước đó.
* **layers.MaxPool2D()**: Lớp tối đa hóa (max pooling) với kích thước cửa sổ 2x2. Lớp này giúp giảm kích thước đầu ra từ các lớp tích chập trước đó.
* Tiếp theo là các lớp tích chập và lớp tối đa hóa lặp lại để trích xuất đặc trưng và giảm kích thước của đầu ra.
* **layers.Flatten()**: Lớp phẳng hóa (flatten) để chuyển đổi đầu ra từ một ma trận 2D thành một vector 1D. Điều này cho phép đầu ra của mạng được đưa vào các lớp kết nối đầy đủ (fully connected layers).
* **layers.Dense(num\_classes, activation="softmax")**: Lớp kết nối đầy đủ với **num\_classes** đơn vị đầu ra và hàm kích hoạt softmax. Đây là lớp đầu ra của mô hình và được sử dụng để phân loại các lớp khác nhau trong bài toán.

Sau khi định nghĩa mô hình, nó được biên dịch bằng cách sử dụng phương thức compile với các tham số như sau:

* **loss="categorical\_crossentropy"**: Hàm mất mát được sử dụng trong quá trình huấn luyện là categorical crossentropy, phù hợp với bài toán phân loại nhiều lớp.
* **optimizer="adam"**: Thuật toán tối ưu hóa được sử dụng là Adam, một phương pháp tối ưu phổ biến và hiệu quả cho huấn luyện mạng nơ-ron.
* **metrics=["accuracy"]**: Đánh giá hiệu suất mô hình bằng chỉ số độ chính xác trong quá trình huấn luyện và kiểm tra.

Đây là một mô hình CNN đơn giản có một số lớp tích chập và lớp kết nối đầy đủ, thích hợp cho các bài toán phân loại ảnh.

A screen shot of a computer program

Description automatically generated with low confidence

Khi train dữ liệu mô hình sử dụng đối số đầu vào là tập train và test đã được tiền xử lý chuẩn hóa cùng với 1 số tham số như sau:

* **steps\_per\_epoch=len(train\_data)**: Đối số **steps\_per\_epoch** xác định số lượng bước (steps) mà mô hình sẽ thực hiện trong mỗi epoch. Trong trường hợp này, nó được đặt bằng độ dài của đối tượng **train\_data**, tức là số lượng lô dữ liệu huấn luyện.
* **epochs=5**: Đối số **epochs** xác định số lượng epoch mà mô hình sẽ được huấn luyện. Trong trường hợp này, mô hình sẽ được huấn luyện trong 5 epoch.
* **validation\_steps=len(test\_data)**: Đối số **validation\_steps** xác định số lượng bước (steps) mà mô hình sẽ thực hiện trong quá trình kiểm tra (validation). Trong trường hợp này, nó được đặt bằng độ dài của đối tượng **test\_data**, tức là số lượng lô dữ liệu kiểm tra.
  1. Phân tích và so sánh
     1. Mô hình SVM

A screenshot of a computer program

Description automatically generated with medium confidence

Sau khi tìm được bộ tham số tốt nhất cho mô hình, ta tiến hành đánh giá mô hình với accuraty\_score. Kết quả thu được độ chính xác là 69.2%

A picture containing text, graphic design, comic, art

Description automatically generated

Kết quả thực nghiệm khi kiểm thử trên tập data mới, model SVM cho kết quả sai ở hình thứ 1 và thứ 4 (nhãn đúng là AI-generated và Photographic)

* + 1. Mô hình CNN

A screenshot of a computer screen

Description automatically generated with low confidence

Dựa trên kết quả về quá trình huấn luyện, ta có thể rút ra một số nhận xét sau:

Mất mát (loss): Từ epoch đầu tiên đến epoch cuối cùng, mất mát trung bình trong quá trình huấn luyện giảm dần, cho thấy mô hình đang học cách tối ưu hóa các tham số để giảm thiểu mất mát. Điều này cho thấy mô hình đang tiến bộ trong việc học từ dữ liệu huấn luyện.

A picture containing plot, line, diagram, screenshot

Description automatically generated

Độ chính xác (accuracy): Từ epoch đầu tiên đến epoch cuối cùng, độ chính xác trung bình trong quá trình huấn luyện tăng dần. Điều này cho thấy mô hình đang học cách phân loại đúng các mẫu huấn luyện. Độ chính xác trên tập kiểm tra (validation) cũng tăng dần theo các epoch, cho thấy mô hình không bị quá khớp (overfitting) và có khả năng tổng quát hóa tốt trên dữ liệu mới.

Thời gian huấn luyện: Mỗi epoch mất khoảng 265 giây để hoàn thành, với mỗi bước mất trung bình 2 giây. Thời gian huấn luyện có thể khá lớn đối với số lượng bước và kích thước dữ liệu lớn. Có thể xem xét việc sử dụng kỹ thuật tăng tốc huấn luyện như sử dụng GPU hoặc phân tán huấn luyện để giảm thời gian.

Đánh giá chung: Sau 5 epoch, mô hình đạt độ chính xác trung bình khoảng 86.22% trên tập kiểm tra (validation). Điều này cho thấy mô hình đã học cách phân loại đúng ảnh mới một cách tương đối tốt sau quá trình huấn luyện.

A picture containing chair, furniture, screenshot, ladder

Description automatically generated

A picture containing text, screenshot, human face, art

Description automatically generated

A picture containing sketch, text, illustration, line art

Description automatically generated

A picture containing text, water, screenshot, lake

Description automatically generated

Khi kiểm thử trên tập dữ liệu mới, model CNN cho kết quả chính xác ở cả 4 nhãn của ảnh.

* + 1. So sánh 2 mô hình

Khi so sánh mô hình SVM và mô hình CNN, có thể rút ra một số nhận xét sau:

Độ chính xác: Mô hình CNN đạt độ chính xác cao hơn so với mô hình SVM. Điều này có nghĩa là mô hình CNN có khả năng phân loại chính xác hơn trên dữ liệu đã sử dụng để đánh giá.

Đặc trưng học tập: SVM và CNN sử dụng các phương pháp học tập khác nhau để xác định đặc trưng của dữ liệu. SVM tập trung vào việc xây dựng các siêu phẳng tốt nhất để phân tách các lớp, trong khi CNN sử dụng mạng thần kinh nơ-ron tích chập để học các đặc trưng từ dữ liệu. Điều này cho thấy CNN có khả năng học các đặc trưng phức tạp hơn và có thể làm việc tốt hơn trên dữ liệu phức tạp.

Tính linh hoạt: SVM có thể áp dụng cho các bài toán phân loại không gian đặc trưng cao, trong khi CNN thường được sử dụng trong các bài toán liên quan đến hình ảnh và dữ liệu có cấu trúc. Vì vậy, lựa chọn mô hình phụ thuộc vào loại dữ liệu và yêu cầu của bài toán cụ thể.

Đánh đổi giữa hiệu suất và tính tốn kém: SVM thường nhanh hơn trong quá trình huấn luyện và dự đoán so với CNN, nhưng CNN có thể đạt được độ chính xác cao hơn. Do đó, khi đánh giá mô hình, ta cần xem xét cân nhắc giữa hiệu suất và tính tốn kém của mô hình trong bối cảnh cụ thể.

Tóm lại, mô hình CNN có xu hướng đạt độ chính xác cao hơn so với SVM trong nhiều bài toán phân loại, đặc biệt là đối với dữ liệu có cấu trúc như hình ảnh. Tuy nhiên, việc lựa chọn mô hình phụ thuộc vào loại dữ liệu và yêu cầu của bài toán cụ thể.

1. TỔNG KẾT
   1. Kết quả đạt được
      1. Mô hình SVM

SVM (Support Vector Machines) là một thuật toán học máy phân lớp, được áp dụng để nhận diện nguồn gốc của ảnh.

Đầu tiên, chúng ta cần xây dựng một tập dữ liệu huấn luyện gồm các ảnh có nhãn (label) cho từng nguồn gốc ảnh mong muốn (tranh vẽ, ảnh kỹ thuật số, ảnh vẽ bởi AI và ảnh chụp).

Tiếp theo, chúng ta trích xuất các đặc trưng từ các ảnh trong tập dữ liệu huấn luyện. Các đặc trưng này có thể bao gồm các thông tin màu sắc, cạnh, hình dạng, hoặc các đặc điểm khác.

Sau đó, chúng ta đưa các đặc trưng đã trích xuất vào mô hình SVM để huấn luyện. SVM sẽ học cách phân loại các nguồn gốc ảnh dựa trên các đặc trưng này.

Sau khi huấn luyện xong, chúng ta có thể sử dụng mô hình SVM đã được huấn luyện để dự đoán nguồn gốc của các ảnh mới.

* + 1. Mô hình CNN

CNN (Convolutional Neural Network) là một mạng nơ-ron sử dụng cấu trúc đặc biệt giúp xử lý ảnh hiệu quả.

Tương tự như SVM, chúng ta cần xây dựng một tập dữ liệu huấn luyện gồm các ảnh có nhãn cho từng nguồn gốc ảnh mong muốn.

Mô hình CNN sẽ bao gồm một hoặc nhiều lớp tích chập (convolutional layers) để tự động trích xuất các đặc trưng từ ảnh. Sau đó, các đặc trưng này sẽ được đưa vào các lớp kết nối đầy đủ (fully connected layers) để phân loại ảnh.

Quá trình huấn luyện mô hình CNN bao gồm việc tối ưu hóa các trọng số của mạng nơ-ron dựa trên độ sai khác giữa nhãn thực tế và dự đoán của mô hình.

Khi mô hình CNN đã được huấn luyện, chúng ta có thể sử dụng nó để dự đoán nguồn gốc của các ảnh mới.

* 1. Hướng phát triển

Để mở rộng mô hình cũng như tang độ chính xác cho mô hình, có thể sử dụng các cách sau:

Thu thập dataset đa dạng, ở nhiều chủ đề hơn

Trích xuất các đặc trưng của hình ảnh

Tinh chỉnh các tham số của mô hình (SVM), tinh chỉnh cấu trúc các lớp (CNN)

Ứng dụng vào một hệ thống thực tế

# DANH MỤC TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | *"Convolutional neural network." Wikipedia. [Truy cập vào ngày 25 tháng 5 năm 2023]. Truy cập từ:* [*https://en.wikipedia.org/wiki/Convolutional\_neural\_network*](https://en.wikipedia.org/wiki/Convolutional_neural_network) |
| [2] | *"Thuật toán CNN - Convolutional Neural Network". TopDev. [Truy cập vào ngày 25 tháng 5 năm 2023]. Truy cập từ:* [*https://topdev.vn/blog/thuat-toan-cnn-convolutional-neural-network/*](https://topdev.vn/blog/thuat-toan-cnn-convolutional-neural-network/) |
| [3] | *"Support vector machine." Wikipedia. [Truy cập vào ngày 25 tháng 5 năm 2023]. Truy cập từ:* [*https://en.wikipedia.org/wiki/Support\_vector\_machine*](https://en.wikipedia.org/wiki/Support_vector_machine)*.* |
| [4] | *"SVM Multiclass Classification." Baeldung. [Truy cập vào ngày 25 tháng 5 năm 2023]. Truy cập từ:* [*https://www.baeldung.com/cs/svm-multiclass-classification*](https://www.baeldung.com/cs/svm-multiclass-classification)*.* |