**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CƠ SỞ TRÍ TUỆ NHÂN TẠO**

**ĐỀ TÀI: XÂY DỰNG HỆ THỐNG NHẬN DẠNG TIỀN VIỆT NAM**

**Giảng viên hướng dẫn: ĐẶNG NHƯ PHÚ**

**Sinh viên thực hiện: TRƯƠNG MINH ĐỈNH**

**MSSV: 2100009776**

**Khoá: 21**

**Ngành/ chuyên ngành: TRÍ TUỆ NHÂN TẠO**

TP. HCM, tháng 09 năm 2023

**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CƠ SỞ TRÍ TUỆ NHÂN TẠO**

**ĐỀ TÀI: XÂY DỰNG HỆ THỐNG NHẬN DẠNG TIỀN VIỆT NAM**

**Giảng viên hướng dẫn: ĐẶNG NHƯ PHÚ**

**Sinh viên thực hiện: TRƯƠNG MINH ĐỈNH**

**MSSV: 2100009776**

**Khoá: 21**

**Ngành/ chuyên ngành: TRÍ TUỆ NHÂN TẠO**

TP. HCM, tháng 09 năm 2023

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH  **KHOA CÔNG NGHỆ THÔNG TIN**  🙜 🙜 🙝 | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  **Độc lập – Tự do – Hạnh phúc**  🙜 🙜 🙝 |

# NHIỆM VỤ ĐỒ ÁN MÔN HỌC

Họ và tên: **TRƯƠNG MINH ĐỈNH** MSSV: **2100009776**

Chuyên ngành: **Trí tuệ nhân tạo**

Lớp: **21DHT2B**

Email: **minhdinh05092003@gmail.com**

SĐT: **0369668926**

Tên đề tài: Xây dựng hệ thống nhận dạng tiền Việt Nam

Giảng viên giảng dạy: **ThS. Đặng Như Phú**

Thời gian thực hiện: **30/06/2023** **đến 01/09/2023**

Nhiệm vụ/nội dung (mô tả chi tiết nội dung, yêu cầu, phương pháp… ):

## Mục tiêu

* Tìm hiểu và áp dụng các kỹ thuật cơ bản của Trí tuệ nhân tạo để dự đoán dựa trên các yếu tố của đầu vào.
* Hiểu được quy trình xây dựng mô hình, huấn luyện và đánh giá mô hình trong Trí tuệ nhân tạo.

## Phạm vi

* Thu thập dữ liệu liên quan đến xây dựng hệ thống nhận dạng tiền Việt Nam
* Tiến hành tiền xử lý dữ liệu để chuẩn bị cho việc huấn luyện mô hình.
* Xây dựng mô hình dự đoán bằng các phương pháp học máy cơ bản (ví dụ: hồi quy tuyến tính).

## Yêu cầu

1. Thu thập dữ liệu và tiền xử lý:

* Thu thập tập dữ liệu và các yếu tố liên quan đến xây dựng hệ thống nhận dạng tiền Việt Nam .
* Tiến hành tiền xử lý dữ liệu để loại bỏ nhiễu và chuẩn hóa.

2. Xây dựng mô hình dự đoán:

* Xây dựng mô hình học máy cơ bản (ví dụ: hồi quy tuyến tính) để dự đoán dựa trên các yếu tố đã chọn.

3. Huấn luyện và đánh giá mô hình:

* Tách tập dữ liệu thành tập huấn luyện và tập kiểm tra.
* Huấn luyện mô hình trên tập huấn luyện và đánh giá hiệu suất trên tập kiểm tra.

4. Thực hiện ứng dụng:

* Xây dựng ứng dụng đơn giản cho phép người dùng nhập thông tin và dự đoán kết quả.

5. Báo cáo đồ án:

* Viết báo cáo về quy trình thực hiện, kết quả đạt được và đánh giá mô hình.
* Chuẩn bị thuyết trình để trình bày kết quả đồ án.

## Tiến độ dự kiến

* Tuần 1-2: Nghiên cứu lý thuyết và thu thập dữ liệu.
* Tuần 3-4: Tiền xử lý dữ liệu và xây dựng mô hình dự đoán.
* Tuần 5-6: Huấn luyện và đánh giá mô hình.
* Tuần 7-8: Xây dựng ứng dụng và tối ưu hóa.
* Tuần 9-10: Hoàn thiện báo cáo đồ án và nội dung thuyết trình.

**Nội dung và yêu cầu đã được thông qua Bộ môn.**

*Tp.HCM, ngày 26 tháng 9 năm 2023*

|  |  |
| --- | --- |
| **TRƯỞNG BỘ MÔN**  *(Ký và ghi rõ họ tên)*  **ThS. Đặng Như Phú** | **GIẢNG VIÊN GIẢNG DẠY**  *(Ký và ghi rõ họ tên)*  **ThS. Đặng Như Phú** |

# LỜI CẢM ƠN

Kính thầy Đặng Như Phú,

Em xin gửi lời biết ơn sâu sắc và lòng tôn kính đến thầy về sự hỗ trợ và sự hướng dẫn xuất sắc của thầy trong quá trình thực hiện đồ án môn học "Đồ án cơ sở trí tuệ nhân tạo" tại trường Đại học Nguyễn Tất Thành. Sự tận tâm và kiến thức sâu rộng của thầy đã là nguồn động viên và cổ vũ vô cùng quý báu đối với em.

Trong suốt khoảng thời gian em được làm việc cùng thầy, em đã học được không chỉ về kiến thức chuyên ngành mà còn về sự cống hiến và đam mê trong công việc. Thầy đã luôn sẵn sàng chia sẻ kiến thức, giải đáp mọi câu hỏi của em và tạo điều kiện để em có thể phát triển tốt nhất khả năng của mình.

Sự hỗ trợ và sự tôn trọng mà thầy dành cho sinh viên như em thật sự là nguồn động viên không giới hạn. Thầy không chỉ giúp em hiểu rõ hơn về trí tuệ nhân tạo mà còn khuyến khích em khám phá và đặt ra những câu hỏi quan trọng, giúp em phát triển tư duy logic và khả năng giải quyết vấn đề.

Em biết ơn thầy vì tất cả những kiến thức và kỹ năng quý báu mà em đã học được dưới sự hướng dẫn của thầy. Sự tận tâm và lòng nhiệt thành của thầy đã góp phần quan trọng vào sự thành công của em trong đồ án này.

Chúng em xin gửi lời cảm ơn chân thành nhất đến thầy Đặng Như Phú và mong rằng thầy sẽ tiếp tục truyền đạt sự đam mê và kiến thức đỉnh cao của mình cho thế hệ sinh viên tiếp theo. Thầy là nguồn cảm hứng lớn lao cho chúng em.

Cảm ơn thầy một lần nữa và chúc thầy luôn được sức khỏe, hạnh phúc và thành công trong mọi sự nghiệp.

Trân trọng,

Trương Minh Đỉnh

# LỜI MỞ ĐẦU

Tiền tệ là một phần không thể thiếu của cuộc sống hàng ngày của chúng ta và đóng vai trò quan trọng trong hệ thống tài chính toàn cầu. Việc đảm bảo tính toàn vẹn và an toàn của tiền tệ là một ưu tiên hàng đầu. Trong bối cảnh này, việc phát triển và triển khai các hệ thống nhận diện tiền tự động đã trở thành một thách thức quan trọng trong lĩnh vực công nghệ thông tin và trí tuệ nhân tạo.

Đồ án này ra đời với mục tiêu tạo ra một hệ thống nhận diện tiền tự động, sử dụng công nghệ xử lý hình ảnh và trí tuệ nhân tạo, nhằm giúp nhận diện và phân loại các loại tiền tệ một cách nhanh chóng và chính xác. Việc phát triển một hệ thống nhận diện tiền tự động không chỉ đòi hỏi sự hiểu biết về lĩnh vực xử lý ảnh và máy học mà còn đòi hỏi sự kết hợp giữa kiến thức chuyên môn và sáng tạo công nghệ.

Trong tình hình mà công nghệ ngày càng phát triển, việc sử dụng trí tuệ nhân tạo để giúp máy tính "nhìn thấy" và "nhận biết" tiền tệ đã mở ra nhiều cơ hội mới trong việc tăng cường hiệu suất và bảo mật trong các giao dịch tài chính, đặc biệt là trong lĩnh vực ngân hàng và thương mại điện tử.

Đồ án này sẽ trình bày quá trình phát triển và triển khai hệ thống nhận diện tiền tự động, từ việc thu thập dữ liệu và tiền xử lý hình ảnh đến việc xây dựng mô hình máy học và tích hợp vào ứng dụng thực tế. Chúng tôi hi vọng rằng đồ án này sẽ cung cấp một cái nhìn chi tiết về quá trình phát triển các ứng dụng trí tuệ nhân tạo trong lĩnh vực tài chính và đóng góp vào sự tiến bộ trong việc bảo vệ tính toàn vẹn của tiền tệ.

Em xin bày tỏ lòng biết ơn sâu sắc đến tất cả những người đã hỗ trợ và đóng góp vào việc thực hiện đồ án này, đặc biệt là giảng viên hướng dẫn và các chuyên gia trong lĩnh vực công nghệ thông tin và trí tuệ nhân tạo.

Mong rằng đồ án này sẽ đem lại giá trị và kiến thức hữu ích cho người đọc và là một bước tiến quan trọng trong lĩnh vực nhận diện tiền tự động.

Cảm ơn quý Thầy\Cô và các bạn đã theo dõi và ủng hộ đồ án của em.

Trương Minh Đỉnh

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH  **TRUNG TÂM KHẢO THÍ** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ III NĂM HỌC 2022 - 2023** |

**PHIẾU CHẤM THI TIỂU LUẬN/ĐỒ ÁN**

BM-ChT-11

Môn thi: Đồ án cơ sở trí tuệ nhân tạo Lớp học phần: 21DTH2A

Nhóm sinh viên thực hiện :

1.Trương Minh Đỉnh – 2100009776 Tham gia đóng góp:Tìm kiếm, thu thập thông tin dữ liệu, xây dựng mô hình hệ thống, viết báo cáo đồ án

Ngày thi: 26/09/2023 Phòng thi: L.501

Đề tài tiểu luận/báo cáo của sinh viên : Xây dựng hệ thống nhận diện tiền Việt Nam

Phần đánh giá của giảng viên (căn cứ trên thang rubrics của môn học):

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu chí (theo CĐR HP)** | **Đánh giá của GV** | **Điểm tối đa** | **Điểm đạt được** |
| Cấu trúc của báo cáo |  | 1 |  |
| Nội dung |  |  |  |
| * Các nội dung thành phần |  | 6 |  |
| * Lập luận |  | 2 |  |
| * Kết luận |  | 0.5 |  |
| Trình bày |  | 0.5 |  |
| **TỔNG ĐIỂM** |  | **10** |  |

|  |  |
| --- | --- |
|  | **Giảng viên chấm thi**  *(ký, ghi rõ họ tên)*  **Đặng Như Phú** |

# MỤC LỤC

NHIỆM VỤ ĐỒ ÁN MÔN HỌC [ii](#_Toc139821559)

[LỜI CẢM ƠN v](#_Toc139821559)

[LỜI MỞ ĐẦU vi](#_Toc139821560)

[MỤC LỤC v](#_Toc139821561)iii

[DANH MỤC BẢNG, HÌNH x](#_Toc139821562)

[DANH MỤC HÌNH xi](#_Toc139821563)

[KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT xii](#_Toc139821564)

[CHƯƠNG 1: Cơ sở lý thuyết 1](#_Toc139821565)

[1.1. Trí tuệ nhân tạo 1](#_Toc139821566)

[1.1.1. Định nghĩa và lịch sử 1](#_Toc139821567)

[1.1.2. Học máy và mạng nơ ron 2](#_Toc139821567)

[1.1.3. Ứng dụng của Trí tuệ nhân tạo 3](#_Toc139821567)

[1.1.4. Thách thức và kết luận 4](#_Toc139821567)

[CHƯƠNG 2: Phân tích yêu cầu 6](#_Toc139821568)

[2.1. Nhiệm vụ và yêu cầu của bài toán 6](#_Toc139821569)

[2.1. Phân tích bài toán 6](#_Toc139821569)

[2.1.1. Các bước thực hiện 7](#_Toc139821567)

[2.1.2. Vấn đề về dữ liệu 8](#_Toc139821567)

[CHƯƠNG 3: Xây dựng mô hình 9](#_Toc139821570)

[3.1. Tạo dữ liệu cho bài toán 9](#_Toc139821571)

[3.2. Xử lý dữ liệu ảnh 11](#_Toc139821571)

[3.3. Thiết kế mạng CNN Classify dùng để train 13](#_Toc139821571)

[3.4. Thực hiện augmentation cho dữ liệu 15](#_Toc139821571)

[CHƯƠNG 4: Thực nghiệm mô hình 17](#_Toc139821572)

[4.1. Train model CNN Classify 17](#_Toc139821573)

[4.2. Train model CNN Classify 18](#_Toc139821573)

[CHƯƠNG 5: Kết luận và hướng phát triển 21](#_Toc139821574)

[5.1. Kết quả đạt được 21](#_Toc139821575)

[5.2. Hạn chế của đề tài 22](#_Toc139821576)

[5.3. Hướng phát triển 22](#_Toc139821577)

[TÀI LIỆU THAM KHẢO 24](#_Toc139821578)

# DANH MỤC BẢNG

# DANH MỤC HÌNH

[*Hình 1:* *Trí tuệ nhân tạo* 1](#_Toc139822309)

[*Hình 2:* *Cấu trúc mạng nơ ron* 2](#_Toc139822309)

[*Hình 3: Ứng dụng của lĩnh vực trí tuệ nhân tạo* 3](#_Toc139822309)

[*Hình 4: Ngành trí tuệ nhân tạo* 5](#_Toc139822309)

[*Hình 5: Cấu trục mạng VGG16* 7](#_Toc139822309)

[*Hình 6: Thực hiện augmentation ảnh bằng cách phóng to thu nhỏ* 8](#_Toc139822309)

[*Hình 7: Một số ảnh của class* 11](#_Toc139822309)

[*Hình 8: Cấu trục mạng VGG16* 13](#_Toc139822309)

[*Hình 9: Mô hình sau khi đã được thay đổi* 15](#_Toc139822309)

[*Hình 10: Quá trình thực hiên train huấn luyện tập dữ liệu mô hình* 1](#_Toc139822309)8

[*Hình 11: Kết quả dự đoán khi chạy chương trình* 2](#_Toc139822309)0

[*Hình 12: Quá trình dự đoán* 22](#_Toc139822309)

# KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **Chữ viết tắt** | **Ý nghĩa** |
| AI | Trí tuệ nhân tạo |
| NLP | Xử lý ngôn ngữ tự nhiên |
| HSV | Hue, Saturation, Value |
| NN | Neural Netword: Mạng ne ron |
| FC | Fully Connected: Tầng kết nối đầy đủ |
| CNN | Convolutional Neural Network |
|  |  |
|  |  |

## CHƯƠNG 1

## Cơ sở lý thuyết

## 1.1. Trí tuệ nhân tạo

### 1.1.1.Định nghĩa và lịch sử:

Trí tuệ nhân tạo (AI) là một lĩnh vực nghiên cứu và ứng dụng ngày càng phát triển, được xem là một trong những công nghệ tiên tiến nhất trong thế kỷ 21. Trong phần này, chúng ta sẽ trình bày cơ sở lý thuyết về trí tuệ nhân tạo trong một phạm vi hơn 1000 từ, nhằm hiểu rõ hơn về khái niệm này và cách nó đã và đang tác động đến cuộc sống của chúng ta.

Trí tuệ nhân tạo là khả năng của máy tính hoặc hệ thống máy tính để thực hiện các tác vụ thông minh mà trước đây chỉ có con người có khả năng thực hiện. Các tác vụ này bao gồm nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên, lập kế hoạch và ra quyết định. Lĩnh vực này đã xuất hiện từ những năm 1950 và đã trải qua nhiều giai đoạn phát triển với nhiều thành tựu đáng kể.

A blue background with a blue background with a blue background with a blue background with a blue background with a white text

Description automatically generated with medium confidence

*Hình 1: Trí tuệ nhân tạo*

### 1.1.2.Học máy và mạng nơ-ron:

Máy học là một phân nhánh của trí tuệ nhân tạo mà nó tập trung vào việc xây dựng các thuật toán và mô hình máy tính có khả năng học từ dữ liệu và cải thiện hiệu suất theo thời gian. Mục tiêu chính của máy học là cho phép máy tính tự động hóa quyết định và học từ kinh nghiệm. Các loại máy học:

+ Học giám sát (Supervised Learning): Máy tính được đào tạo bằng dữ liệu đã được gán nhãn (label), với mục tiêu học cách dự đoán kết quả cho dữ liệu mới.

+ Học không giám sát (Unsupervised Learning): Máy tính học từ dữ liệu không có nhãn và cố gắng phát hiện cấu trúc ẩn trong dữ liệu, chẳng hạn như gom nhóm dữ liệu hoặc giảm chiều dữ liệu.

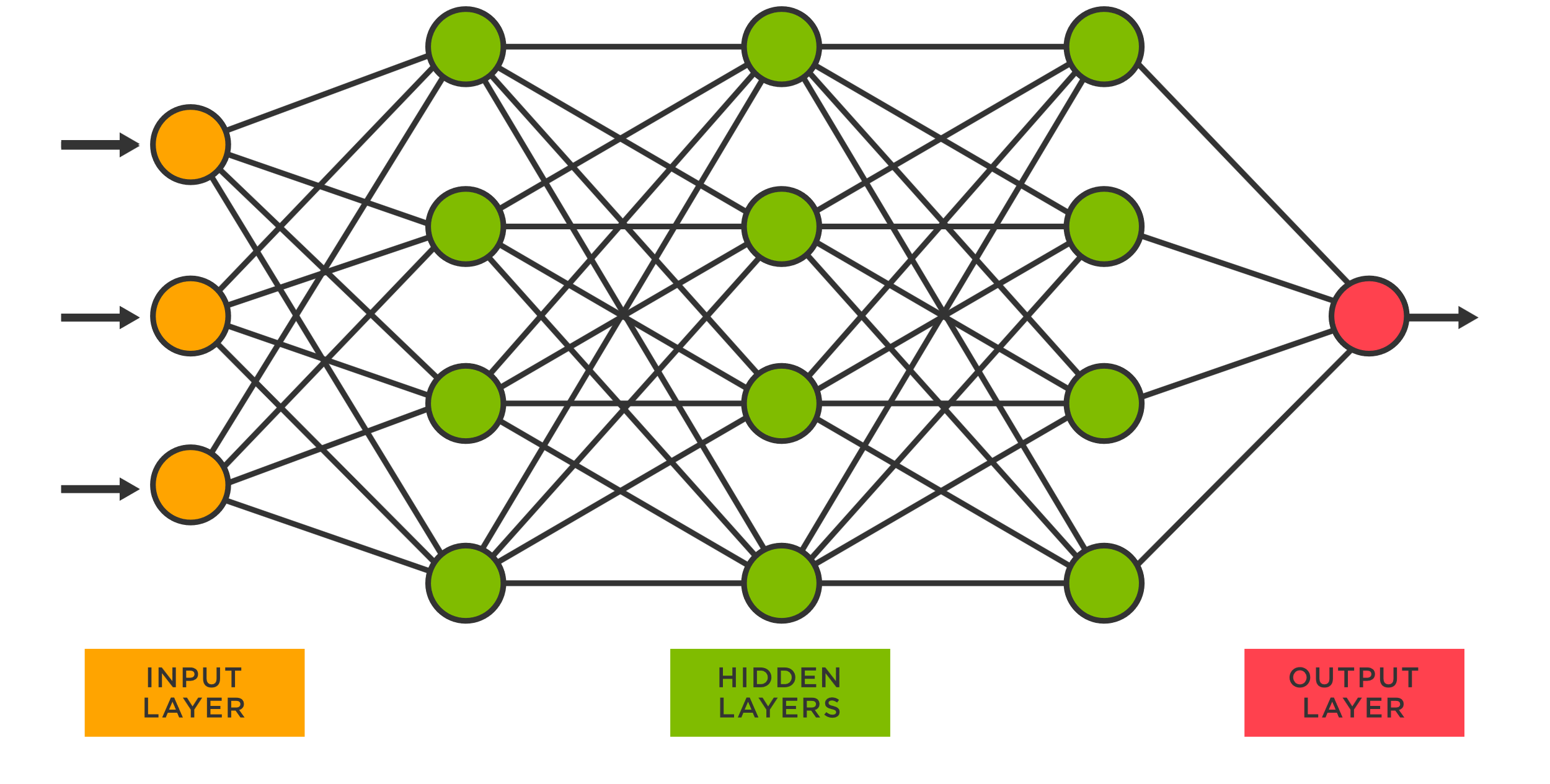
+ Học tăng cường (Reinforcement Learning): Máy tính tương tác với môi trường và học thông qua phản hồi (reward) để tối ưu hóa một nhiệm vụ cụ thể.

Mạng nơ-ron là một mô hình toán học lấy cảm hứng từ cấu trúc não người. Nó bao gồm các nơ-ron nhân tạo (artificial neurons) kết nối với nhau để thực hiện các phép tính phức tạp. Mạng nơ-ron thường được sử dụng trong nhiều ứng dụng máy học, đặc biệt là trong học sâu (Deep Learning). Cấu trúc của Mạng Nơ-ron:

+ Input Layer: Nhận dữ liệu đầu vào và truyền chúng qua mạng.

+ Hidden Layers: Là các tầng nơ-ron ẩn giữa input layer và output layer. Chúng thực hiện các phép tính phức tạp và học cách biểu diễn dữ liệu.

+ Output Layer: Trả về kết quả cuối cùng của mạng, thường là dự đoán hoặc phân loại.



*Hình 2: Cấu trúc mạng nơ ron*

Trong kết luận, máy học và mạng nơ-ron là hai khái niệm quan trọng trong lĩnh vực trí tuệ nhân tạo. Chúng đóng vai trò quan trọng trong việc phát triển các ứng dụng thông minh và có tiềm năng biến đổi nhiều khía cạnh của cuộc sống và công nghiệp.

### 1.1.3. Ứng dụng của Trí tuệ nhân tạo:

Trí tuệ nhân tạo đã và đang có ảnh hưởng lớn đến nhiều khía cạnh của cuộc sống chúng ta. Dưới đây là một số ứng dụng quan trọng của AI:

+ Xử lý ngôn ngữ tự nhiên (NLP): NLP cho phép máy tính hiểu và tương tác với ngôn ngữ con người. Điều này đã tạo ra các ứng dụng như trợ lí ảo, dịch thuật tự động và phân tích ngôn ngữ tự động.

+ Nhận diện hình ảnh: Các hệ thống AI có khả năng nhận diện và phân loại hình ảnh, từ việc nhận dạng khuôn mặt cho đến nhận biết vật thể trong ảnh và phát hiện các biểu đồ y tế trên hình X-quang.

+ Tự lái và xe tự hành: AI đã giúp phát triển xe tự hành, làm giảm tai nạn giao thông và cải thiện tính tiện dụng trong việc đi lại.

+ Dự đoán và phân tích dữ liệu: Các thuật toán AI có khả năng phân tích dữ liệu lớn và dự đoán xu hướng, giúp cho các lĩnh vực như tài chính, y tế và tiếp thị.

A diagram of different medical devices

Description automatically generated with medium confidence

*Hình 3: Ứng dụng của lĩnh vực trí tuệ nhân tạo*

### 1.1.4. Thách thức và kết luận:

Ngành trí tuệ nhân tạo (AI) đang đối mặt với nhiều thách thức quan trọng và đa dạng, bao gồm:

- Hiểu biết và Giám sát: Cách AI hiểu và học từ dữ liệu vẫn chưa rõ ràng. Thậm chí khi mô hình AI thực hiện công việc tốt, chúng ta cũng không biết chính xác cách chúng đã thực hiện nó. Điều này làm tăng rủi ro liên quan đến hệ thống AI ra quyết định sai lầm hoặc không minh bạch.

- Trí tuệ giả mạo (AI Fakery): AI có thể tạo ra thông tin giả mạo, video giả mạo, hoặc âm thanh giả mạo dễ dàng. Điều này có thể dẫn đến việc lan truyền tin giả mạo và đe dọa tính minh bạch và tin cậy của thông tin trực tuyến.

- Sự riêng tư và Bảo mật: Thu thập và sử dụng dữ liệu cá nhân đang trở thành một vấn đề nhạy cảm. AI cần phải đảm bảo sự riêng tư và bảo mật của dữ liệu người dùng trong quá trình hoạt động.

- Khả năng quyết định đạo đức: AI đôi khi phải ra quyết định có tính đạo đức, như trong trường hợp xe tự hành phải quyết định giữa cuộc va chạm hoặc thương vong. Điều này đặt ra câu hỏi về trách nhiệm và đạo đức trong thiết kế và triển khai AI.

- Thất nghiệp và Thay thế công việc: Sự tự động hóa bằng AI có thể dẫn đến việc mất việc làm cho nhiều người trong một số ngành công nghiệp, đặc biệt là trong sản xuất và dịch vụ khách hàng.

- Biased và Unfair AI: AI có thể phản ánh sự thiên vị và không công bằng trong dữ liệu huấn luyện, dẫn đến các quyết định không công bằng hoặc độc đoán. Điều này đặt ra câu hỏi về công bằng và đạo đức trong thiết kế và triển khai AI.

- Nhận thức về AI và Giáo dục: Sự hiểu biết và nhận thức về AI trong cộng đồng cần được nâng cao. Công chúng cần biết cách sử dụng AI một cách an toàn và hiểu rõ về tiềm năng và giới hạn của nó.

- Quản lý và Quy định: Cần có quy định và quản lý rõ ràng về việc sử dụng AI để đảm bảo rằng nó được áp dụng một cách đúng đắn và an toàn.

- Tài nguyên tính toán và Năng suất: Đào tạo và triển khai các mô hình AI cần nhiều tài nguyên tính toán, và điều này có thể tạo ra vấn đề về sự cạnh tranh và tiêu thụ năng lượng.

- Hạn chế trong việc tự học: Hiện tại, AI vẫn có hạn chế trong việc tự học và tự hiểu, và nó còn rất phụ thuộc vào dữ liệu và sự giám sát của con người.

Các thách thức này yêu cầu sự cộng tác giữa cộng đồng AI, ngành công nghiệp, và các cơ quan quản lý để đảm bảo rằng AI được phát triển và sử dụng một cách bền vững và đạo đức.

Trí tuệ nhân tạo đang chuyển đổi cách chúng ta sống và làm việc. Được hỗ trợ bởi học máy và mạng nơ-ron, AI đang có ứng dụng mạnh mẽ trong nhiều lĩnh vực khác nhau. Tuy nhiên, cần phải cân nhắc kỹ lưỡng về các vấn đề liên quan đến đạo đức và an ninh thông tin để đảm bảo sự phát triển bền vững và tích cực của trí tuệ nhân tạo trong tương lai.

A person using a computer

Description automatically generated

*Hình 4: Ngành trí tuệ nhân tạo*

# CHƯƠNG 2

# Phân tích yêu cầu

## 2.1. Nhiệm vụ và yêu cầu của bài toán:

Nhiệm vụ của bài toán là: khi chúng ta đưa các tờ tiền có mệnh giá lần lượt là 10.000VNĐ, 20.000VNĐ, 50.000VNĐ và không cầm tiền vào camera, hệ thống phải nhận diện các mệnh giá của từng tờ tiền lẫn trường hợp không cầm tiền và đưa ra kết quả đã phân tích cho người dùng. Bài này có nhiều cách làm như OpenCV, tách màu HSV nhưng trong bài toán này ta sẽ áp dụng mô hình mạng kinh điển VGG16 và ta trên cơ sở là một bài toán đơn giản để học tập và tham khảo cho sinh viên cuối năm hai thì ta sẽ thực hiện một số yêu cầu sau để có thể tối ưu bài toán:

- Tự tạo dữ liệu cho bài toán: Vì là bài toán nhận dạng tiền nên dữ liệu khá nhạy cảm và cũng theo yêu cầu học tập nên ta sẽ thực hiện tạo tập dữ liệu cho bài toán bằng cách đọc 1 tập dữ liệu từ camera để đưa vào sử dụng.

- Augment dữ liệu tránh Overfit cho model: Dữ liệu đưa vào khá hạn chế nên ta sẽ áp dụng 1 số các phương pháp để làm đa dạng dữ liệu cho mô hình bằng cách xoay ảnh, phòng ảnh, lật ảnh, xoay ngang dọc…

- Cách “thiết kế” mạng NN cho bài toán dựa trên khung nền của mạng VGG16: Ta áp dụng cấu trúc có sẵn của mạng VGG16 và sẽ chỉnh sửa thay thế cấu trúc phần đầu ra để phù hợp với yêu cầu bài toán.

A diagram of a layer of layers

Description automatically generated

*Hình 5: Cấu trục mạng VGG16*

## 2.2. Phân tích bài toán:

## 2.2.1.Các bước thực hiện:

Bài toán thực hiện sẽ gồm các bước như sau:

- Tạo dữ liệu bằng cách đọc ảnh các tờ tiền từ camera. Thực hiện đọc 1000 ảnh từ camera cho từng lớp “10000”, “20000”, “50000” và không cầm tiền

*Hình 5:*

- Xử lý ảnh : Thực hiện convert và resize ảnh về kích thước phù hợp đạt một tiêu chuẩn nhất định. Cụ thể convert các nhãn (10000, 20000, 50000, không cầm tiền) về dạng one-hit, resize ảnh về 128x128 đúng tiêu chuẩn đầu vào của mô hình đã được xây dựng

- Thiết kế mạng NN với đầu vào là ảnh (128,128,3), đưa vào mạng VGG16 và đầu ra của VGG16 sẽ dùng để đưa vào 1 mạng NN nhỏ kết thúc bằng 1 lớp Dense và hàm softmax.

- Sử dụng augmentation cho dữ liệu: để dữ liệu được đa dạng mô hình có thể huấn luyện tốt hơn ta thực hiện augmentation bằng nhiều phương pháp khác nhau như phóng to thu nhỏ ảnh ngẫu nhiên, dịch trái dịch phải, dịch lên dịch xuống, tăng giảm độ sáng, lật ảnh, xoay ảnh, làm méo…

A collage of a person holding a paper money

Description automatically generated

*Hình 6: Thực hiện augmentation ảnh bằng cách phóng to thu nhỏ*

- Đầu ra sẽ là 1 vector softmax chứa các probality p(i) ứng với mỗi class i, chúng ta sẽ in ra giá trị max trong vector đó và chọn đó làm class dự đoán.

## 2.2.2 Vấn đề về dữ liệu:

Bài toán dự kiến làm sample với 3 class là 10000,20000 và 50000. Vậy theo suy nghĩ thông thường thì chỉ cần chuẩn bị dữ liệu cho 3 class và lớp Dense output cuối cùng sẽ là Dense(3).

Tuy nhiên do tổng các p(i) phải = 1 nên giả sử khi chúng ta không cầm đồng tiền nào thì máy sẽ buộc phải dự đoán vào 1 trong 3 class nói trên, dó là điều vô lý. Vì vậy chúng ta sẽ train với 4 class output là “Không tiền”,”10000″,”20000″ và “50000”, output sẽ là Dense(4).

# CHƯƠNG 3

# Xây dựng mô hình

## 3.1. Tạo dữ liệu cho bài toán:

Có bài toán thì có dữ liệu cho trước rất nhiều trên mạng, ví dụ face, ví dụ ảnh đồ vật, ảnh phong cảnh. Tuy nhiên các dữ liệu về tiền, đặc biệt tiền Việt thì tìm mãi chả ra nên phải tự tạo cho mình.

Cách tạo dữ liệu đơn giản là viết một đoạn python đọc liên tục từ camera và save lại vào các thư mục tương ứng ảnh các tờ tiền. Ví dụ trong bài này mình làm sample với 3 class là 10000,20000 và 50000 nhé (các bạn chú ý sửa cái dòng label = “00000” thành class mà mình muốn capture dữ liệu )

*import* numpy *as* np  
*import* cv2  
*import* time  
*import* os  
  
*# Label= "00000": không cầm tiền; "10000": mệnh giá 10k; "20000": mệnh giá 20k; "50000": mệnh giá 50k*label = "50000"  
  
cap = cv2.VideoCapture(0)  
  
*# Biến đếm, để chỉ lưu dữ liệu sau khoảng 60 frame, tránh lúc đầu chưa kịp cầm tiền lên*i = 0  
*while True*:  
 i += 1  
 ret, frame = cap.read()  
 *if not* ret:  
 *continue* frame = cv2.resize(frame, dsize=*None*, fx=1, fy=1)  
 cv2.imshow('frame', frame)

*if* i >= 60 *and* i <= 1060:  
 print("Số ảnh capture = ", i - 60) *if not* os.path.exists('data/' + str(label)):  
 os.mkdir('data/' + str(label))  
  
 cv2.imwrite('data/' + str(label) + "/" + str(i) + ".png", frame)  
  
 *if* cv2.waitKey(1) & 0xFF == ord('q'):  
 *break*

cap.release()  
cv2.destroyAllWindows()

Sau khi chạy thành công 4 class, chúng ta sẽ có thư mục data với các subfolders như sau (chú ý tạo thư mục data trước khi chạy nếu không sẽ bị lỗi):

A number on a black background

Description automatically generated

Đảm bảo mỗi class có số ảnh tương đồng nhau, bài toán này ta sẽ lấy 1000 ảnh cho mỗi class. Mình có in số ảnh capture được ra màn hình và kết thức khi đủ 1000 ảnh.

Ví dụ 1 số ảnh data với mỗi class:



*Hình 7: Một số ảnh của class*

Như vậy là xong phần tạo dữ liệu. Bây giờ chúng ta sẽ tiến hành xử lý dữ liệu đó và ghi vào file dữ liệu pickle.

## 3.2. Xử lý dữ liệu ảnh:

Chú ý như sau, trong quá trình chúng ta train model, thử nghiệm model thì sẽ phải chạy chương trình rất nhiều lần để debug, sửa lỗi… Như vậy mà mỗi lần chạy lại phải tìm hết file và thư mục trên thì sẽ khá lâu nên chúng ta sẽ đọc 1 lần và thực hiện:

- Convert nhãn (là cái mớ 00000,10000,20000….) thành one-hot

- Resize ảnh về 128×128

và lưu vào file pickle để lần sau load cho tiện. Dưới đây là đoạn mã code function save\_data() để xử lý ảnh và lưu vào file pix.data nên chỉ cần chạy đoạn mã một lần và sử dụng file pix.data đã được sinh ra.

*def* save\_data(raw\_folder=raw\_folder):  
  
 dest\_size = (128, 128)  
 print("Bắt đầu xử lý ảnh...")  
  
 pixels = []  
 labels = []  
  
 *# Lặp qua các folder con trong thư mục raw  
 for* folder *in* listdir(raw\_folder):  
 *if* folder!='.DS\_Store':  
 print("Folder=",folder)  
 *# Lặp qua các file trong từng thư mục chứa các em  
 for* file *in* listdir(raw\_folder + folder):  
 *if* file!='.DS\_Store':  
 print("File=", file)  
 pixels.append( cv2.resize(cv2.imread(raw\_folder + folder +"/" + file),dsize=(128,128)))  
 labels.append( folder)  
  
 pixels = np.array(pixels)  
 labels = np.array(labels)*#.reshape(-1,1)  
  
 from* sklearn.preprocessing *import* LabelBinarizer  
 encoder = LabelBinarizer()  
 labels = encoder.fit\_transform(labels)  
 print(labels)  
  
 file = open('pix.data', 'wb')  
 *# dump information to that file* pickle.dump((pixels,labels), file)  
 *# close the file* file.close()  
  
 *return*

Sau khi lưu xong thì chúng ta không cần quan tâm đến cái folder data kia nữa mà chỉ quan tâm đến file pix.data mới sinh ra. Nó chứa full cả ảnh và labels rồi.

Khi nào cần dùng thì load lên và sử dụng lại. Dưới đây là mã code function load\_data() để load dữ liệu cho bài toán sau khi đã xử lý dữ liệu ảnh ở phía bên trên

*def* load\_data():  
 file = open('pix.data', 'rb')  
  
 *# dump information to that file* (pixels, labels) = pickle.load(file)  
  
 *# close the file* file.close()  
  
 print(pixels.shape)  
 print(labels.shape)  
  
  
 *return* pixels, labels

Thực hiện gọi các function save\_data() và load\_data() để thực hiện xử lý dữ liệu ảnh và thực hiện phân chia tập dữ liệu trainset và testset bằng hàm train\_test\_split() của thư viện sklearn.

save\_data()  
X,y = load\_data()  
X\_train, X\_test, y\_train, y\_test = train\_test\_split( X, y, test\_size=0.2, random\_state=100)

## 3.3. Thiết kế mạng CNN Classify dùng để train:

Chúng ta đã chuẩn bị dữ liệu xong, xử lý xong và đây là phần quan trọng nhất – thiết kế mạng để train. Đây là mô hình quan trọng cũng như cốt lõi không thể thiếu của bài toán khi thực hiện xây dựng mô hình vì đây là phần xử lý dữ liệu chạy bài toán tính toán các số liệu để đưa kết quả đầu ra cho người dùng.

Chúng ta đã biết cấu trúc của một mạng CNN, cụ thể là VGG16 sẽ có dạng như sau:

A diagram of a layer of layers

Description automatically generated

*Hình 8: Cấu trục mạng VGG16*

Nhìn vào cấu trúc VGG16 sẽ gồm 2 phần, phần màu cam là trích đặt trưng của ảnh còn phần màu tím sẽ là các lớp FC để classify. Nhưng classify ở đây là dùng cho mục đích của mạng VGG16 có sẵn và theo yêu cầu bài toán thì chúng ta sẽ thiết kế lại phần này cho phù hợp với bài toán nhận dạng tiền

Chúng ta sẽ bỏ đi phần tím classifier bằng cách khai báo include\_top=False khi implement cái mạng này:

model\_vgg16\_conv = VGG16(weights='imagenet',include\_top=*False*)

Rồi bây giờ chúng ta thực hiện ghép nối cái FC của chúng ta vào bằng đoạn lệnh:

*for* layer *in* model\_vgg16\_conv.layers:  
 layer.trainable = *False  
  
# Tao model*input = Input(shape=(128, 128, 3), name='image\_input')  
output\_vgg16\_conv = model\_vgg16\_conv(input)  
  
*# Them cac layer FC va Dropout*x = Flatten(name='flatten')(output\_vgg16\_conv)  
x = Dense(4096, activation='relu', name='fc1')(x)  
x = Dropout(0.5)(x)  
x = Dense(4096, activation='relu', name='fc2')(x)  
x = Dropout(0.5)(x)  
x = Dense(4, activation='softmax', name='predictions')(x)  
  
*# Compile*my\_model = Model(inputs=input, outputs=x)  
my\_model.compile(loss='categorical\_crossentropy', optimizer='adam', metrics=['accuracy'])  
  
*return* my\_model

Chú ý ở đây phải có đoạn lệnh # Đóng băng các layers, đoạn này nghĩa là khi train ta chỉ điều chỉnh weights của phần FC thêm vào

Chú ý layers cuối cùng là Dense(4) với activation softmax như chúng ta phân tích ở trên. Mạng cuối cùng sẽ như sau, phần màu xanh lá cây là lớp FC mà chúng ta thêm vào, output sẽ là 1 vector chứ probality từng class.

A diagram of a diagram of a diagram

Description automatically generated with medium confidence

*Hình 9: Mô hình sau khi đã được thay đổi*

Tất cả các đã mã code trên được tạo trong function get\_model() như sau:

*def* get\_model():  
 model\_vgg16\_conv = VGG16(weights='imagenet', include\_top=*False*)  
  
 *# Dong bang cac layer  
 for* layer *in* model\_vgg16\_conv.layers:  
 layer.trainable = *False  
  
 # Tao model* input = Input(shape=(128, 128, 3), name='image\_input')  
 output\_vgg16\_conv = model\_vgg16\_conv(input)  
  
 *# Them cac layer FC va Dropout* x = Flatten(name='flatten')(output\_vgg16\_conv)  
 x = Dense(4096, activation='relu', name='fc1')(x)  
 x = Dropout(0.5)(x)  
 x = Dense(4096, activation='relu', name='fc2')(x)  
 x = Dropout(0.5)(x)  
 x = Dense(4, activation='softmax', name='predictions')(x)  
  
 *# Compile* my\_model = Model(inputs=input, outputs=x)  
 my\_model.compile(loss='categorical\_crossentropy', optimizer='adam', metrics=['accuracy'])  
  
 *return* my\_model

## 3.4. Thực hiện augmentation cho dữ liệu:

Bây giờ nếu chúng ta sử dụng ngay ảnh nói trên để train cho model CNN Classify thì sẽ bị hiện tượng Overfit vì dữ liệu nhiều nhưng đa phần giống nhau. Dẫn đến train sẽ có chất lượng tốt nhưng khi test sẽ thấy không nhận đúng kết quả cần dự đoán.

Chúng ta sẽ thực hiện augment dữ liệu để làm phong phú hơn dữ liệu, tăng data variance , tăng tính tổng quát cho model bằng ImageDataGenerator của Keras.

Trong bài này chúng ta dựa vào thực tế bài toán nên chỉ sử dụng các phép augment như sau:

*# construct the training image generator for data augmentation*aug = ImageDataGenerator(rotation\_range=20, zoom\_range=0.1,  
 rescale=1./255,  
 width\_shift\_range=0.1,  
 height\_shift\_range=0.1,  
 horizontal\_flip=*True*,  
 brightness\_range=[0.2,1.5], fill\_mode="nearest")  
  
aug\_val = ImageDataGenerator(rescale=1./255)

+ rotation\_range=20: Xoay ảnh một góc 20 độ.

+ zoom\_range=0.1: Thu nhỏ ảnh 10%

+ rescale=1./255: Chỉnh ảnh

+ width\_shift\_range=0.1: dịch ngang qua trái/phải ngẫu nhiên tối da 1/10 độ rộng ảnh

+ height\_shift\_range=0.1: dịch dọc trên/dưới ngẫu nhiên tối đa 1/10 độ cao ảnh

+ horizontal\_flip=True: Xoay theo chiều ngang

+ brightness\_range=[0.2,1.5]: Điều chỉnh độ sáng của ảnh trong khoảng 0.2 đến 1.5

# CHƯƠNG 4

# Thực nghiệm mô hình

## 4.1. Train model CNN Classify:

Phần này ta sẽ thực hiện tiếp phân source trong file train.py.

Nếu như mọi lần chúng ta sử dụng lệnh model.fit để train thì lần này chúng ta sẽ sử dụng model.fit\_generator . Lý do đơn giản chúng ta sinh ra dữ liệu trong quá trình train.

vgghist=vggmodel.fit\_generator(aug.flow(X\_train, y\_train, batch\_size=64),  
 epochs=50,*# steps\_per\_epoch=len(X\_train)//64,* validation\_data=aug.flow(X\_test,y\_test,  
 batch\_size=64),  
 callbacks=callbacks\_list)  
  
vggmodel.save("vggmodel.h5")

Mình cũng đã tạo checkpoint để lưu lại các weights tốt:

filepath="weights-{epoch:02d}-{val\_accuracy:.2f}.hdf5"  
checkpoint = ModelCheckpoint(filepath, monitor='val\_accuracy', verbose=1, save\_best\_only=*True*, mode='max')  
callbacks\_list = [checkpoint]

Sau khi thực hiện quá trình train huấn luyện cho mô hình thì ta sẽ thu được file vggmodel.h5 và file weights-39-097.hdf5 sau khi thực hiện 50 epoch thì tại lần lặp thứ 37 đã đạt được độ chính xác tốt nhất bằng 97%.

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

*Hình 10: Quá trình thực hiên train huấn luyện tập dữ liệu mô hình*

## 4.2. Thử nghiệm model CNN Classify:

Bây giờ các bạn chạy file test.py để kiểm thử model. File này khá đơn giản:

+ Đọc ảnh từ camera

+ Thực hiện resize ảnh, normalize ảnh, chuyển thành tensor và đưa vào model để predict

+ Lấy kết quả đầu ra và hiển thị trên màn hình

Chúng ta sẽ load file weights đã được checkpoint lưu lại ở trên ở đoạn này:

my\_model = get\_model()  
my\_model.load\_weights("weights-39-0.97.hdf5")

Đồng thời, chúng ta chỉ xét các ảnh được nhận dạng có probality > 0.8 và hiển thị lên màn hình .Dưới đây là mã code đọc ảnh từ camera và đưa ra kết quả dự đoán cho người dùng.

*while* (*True*):  
 *# Capture frame-by-frame  
 #* ret, image\_org = cap.read()  
 *if not* ret:  
 *continue* image\_org = cv2.resize(image\_org, dsize=*None*, fx=0.5, fy=0.5)  
 *# Resize* image = image\_org.copy()  
 image = cv2.resize(image, dsize=(128, 128))  
 image = image.astype('float') \* 1. / 255  
 *# Convert to tensor* image = np.expand\_dims(image, axis=0)  
  
 *# Predict* predict = my\_model.predict(image)  
 print("This picture is: ", class\_name[np.argmax(predict[0])], (predict[0]))  
 print(np.max(predict[0], axis=0))  
 *if* (np.max(predict) >= 0.8) *and* (np.argmax(predict[0]) != 0):  
 *# Show image* font = cv2.FONT\_HERSHEY\_SIMPLEX  
 org = (50, 50)  
 fontScale = 1.5  
 color = (0, 255, 0)  
 thickness = 2  
  
 cv2.putText(image\_org, class\_name[np.argmax(predict)], org, font,  
 fontScale, color, thickness, cv2.LINE\_AA)  
  
 cv2.imshow("Picture", image\_org)  
  
 *if* cv2.waitKey(1) & 0xFF == ord('q'):  
 *break  
  
# When everything done, release the capture*cap.release()  
cv2.destroyAllWindows()

A hand holding a paper currency

Description automatically generated

*Hình 11: Kết quả dự đoán khi chạy chương trình*

# CHƯƠNG 5

# Kết luận và hướng phát triển

## 5.1. Kết quả đạt được

Sau khi thực hiện chương trình xây dựng mô hình hệ thống nhận dạng tiền Việt Nam thành công trên tinh thần sử dụng kiến thức cơ bản và phục vụ mục đích học tập học hỏi kinh nghiệm dựa trên một mô hình nhỏ nhưng mang lại nhiều kiến thức bổ ích cho bản thân sinh viên. Qua đó nắm được các khái niệm và kĩ thuật cơ bản như:

- Thu thập và tạo tập dữ liệu cho bài toán.

- Tiếp cận và tái sử dựng các mô hình mạng có sẵn với độ chính xác cao được sử dụng phổ biến trong lĩnh vực trí tuệ nhân tạo.

- Thực hiện điều chỉnh các thông số các lớp mạng của mô hình cụ thể là mô hình mạng VGG16 cho phù hợp với yêu cầu và nhiệm vụ của bài toán đã đặt ra

- Tiếp cận đến mô hình train huấn luyện dữ liệu.

- Thực hiện các kĩ thuật augmentation dữ liệu để đa dạng và phong phú thêm cho tập dữ liệu đầu vào. Tránh trường hợp overfit làm sai lệch giảm độ chính xác của bài toán khi áp dụng vào thực tế.

Qua đó bài tán đã thực hiện dự đoán khi người dùng đưa ảnh các tờ tiền mệnh giá 10000VNĐ, 20000VNĐ, 50000VNĐ và không đưa tiền vào camera thì mô hình đã cho ra kết quả dự đoán khá tin cậy.

A hand holding a paper currency

Description automatically generated

*Hình 12: Quá trình dự đoán*

Sau bài toán trên em đã rút ra cho bản thân rất nhiều kiến thức trong lĩnh vực dự đoán nói riêng và lĩnh vực trí tuệ nhân tạo nói chung từ đó dựa vào các kiến thức đã có trên để thực hiện các bài toán khác với mục đích nâng cao hơn phù hợp với như cầu thực tế đem lại hiệu quả cho người dùng.

## 5.2. Hạn chế của đề tài

Tuy bài toán khá thành công nhưng bài toán cũng gặp không ít vấn đề trong quá trình xây dựng, huấn luyện và chạy bài toán:

- Dữ liệu cho bài toán quá ít và không đa dạng cho quá trình train vì vậy cần thu thập thêm tập dữ liệu hình ảnh gồm nhiều cấu trúc ảnh khác nhau để tăng độ học của bài toán thêm phong phú

- Mô hình dự đoán chưa tối ưu khi mới chỉ có 4 lớp đầu ra vì vậy cần xây dựng thêm các lớp dữ liệu để mở rộng tập dự dữ liệu có thể dữ đoán được.

…

## 5.3. Hướng phát triển

Bài này đưa vào thực tế cần làm thêm nhiều việc như:

- Train thêm các loại tiền khác(Tất cả các loại tiền Việt Nam)

- Predict trong vài frame liên tục và lấy bình quân có trọng số để tăng độ chính xác, tránh nhận nhầm.

- Giới hạn vùng xử lý (ví dụ ở giữa khung hinh) để tăng tốc độ xử lý video…

**DANH MỤC TÀI LIỆU THAM KHẢO**

1. Cao Hoàng Trụ (2014)*, Trí tuệ nhân tạo = thông minh + giải thuật*, Nhà xuất bản Đại học quốc gia TP. Hồ Chí Minh.

2. Dương Anh Tuấn (2018), *Phân tích và thiết kế giải thuật*, NXB Đại học quốc gia TP. Hồ Chí Minh.

3. Trần Minh Quang (2020), *Khai phá dữ liệu và kĩ thuật phân lớp* , NXB Đại học quốc gia TP. Hồ Chí Minh.

4.MiAI.com,*https://www.miai.vn/2020/04/21/nhan-dang-tien-viet-nam-voi-transfer-learning-vgg16-cnn-classify/,* 12/09/2023