

**TRƯỜNG ĐẠI HỌC DUY TÂN**  
**KHOA KHOA HỌC MÁY TÍNH**



**XÂY DỰNG HỆ THỐNG CHĂM SÓC SỨC KHỎE**  
**CHO NGƯỜI TIỂU ĐƯỜNG**

**KHÓA LUẬN CỬ NHÂN CÔNG NGHỆ THÔNG TIN**

**HÀ NGUYỄN PHƯỚC VŨ**  
**VÕ TẤN VĂN**

**ĐÀ NẴNG, 2022**

**TRƯỜNG ĐẠI HỌC DUY TÂN**  
**KHOA KHOA HỌC MÁY TÍNH**



**KHÓA LUẬN TỐT NGHIỆP**  
**XÂY DỰNG HỆ THỐNG CHĂM SÓC SỨC KHỎE**  
**CHO NGƯỜI TIỂU ĐƯỜNG**

**TÀI LIỆU TÓM TẮT**

**GVHD: TS. Lê Thanh Long**

**SVTH:**

- |                              |                      |
|------------------------------|----------------------|
| <b>1. Võ Tấn Văn</b>         | <b>- 24211206496</b> |
| <b>2. Hà Nguyễn Phước Vũ</b> | <b>- 24217200619</b> |

**ĐÀ NẴNG, 2022**

## MỤC LỤC

### CHƯƠNG 1: TỔNG QUAN DỰ ÁN 1

|  |  |    |
|--|--|----|
| 1.1  | Định nghĩa dự án .....                         | 1  |
| 1.2  | Mô tả vấn đề .....                             | 1  |
| 1.3  | Đề xuất giải pháp .....                        | 1  |
| CHƯƠNG 2: PHÂN LOẠI VÀ PHÂN ĐOẠN HÌNH ẢNH .....  |  | 2  |
| 2.1  | Phân loại hình ảnh .....                       | 2  |
| 2.1.1  | Khái niệm .....                                | 2  |
| 2.1.2  | Kỹ thuật phân loại ảnh .....                   | 2  |
| 2.1.3  | Cấu trúc của bài toán phân loại hình ảnh ..... | 5  |
| 2.2  | Phân đoạn hình ảnh .....                       | 6  |
| 2.2.1  | Khái niệm .....                                | 6  |
| 2.2.2  | Cách tiếp cận .....                            | 6  |
| CHƯƠNG 3: MẠNG TÍCH CHẬP .....                   |  | 7  |
| 3.1  | Mạng thần kinh nhân tạo .....                  | 7  |
| 3.2  | Mạng tích chập .....                           | 8  |
| CHƯƠNG 4: XÂY DỰNG HỆ THỐNG .....                |  | 9  |
| 4.1  | Đề xuất kỹ thuật .....                         | 9  |
| 4.2  | Usecase hệ thống .....                         | 9  |
| 4.3  | Product backlog .....                          | 10 |
| 4.4  | Cơ sở dữ liệu .....                            | 11 |
| 4.4.1  | Tổng quan về bảng .....                        | 11 |
| 4.4.2  | Sơ đồ mối quan hệ bảng .....                   | 12 |
| CHƯƠNG 5: XÂY DỰNG MODULE MACHINE LEARNING ..... |  | 13 |
| 5.1  | Vấn đề .....                                   | 13 |
| 5.2  | Phương pháp giải quyết .....                   | 13 |
| 5.3  | Mục tiêu .....                                 | 13 |
| 5.4  | Mô hình thực nghiệm và triển khai .....        | 13 |
| 5.4.1  | Mục tiêu 1 .....                               | 13 |
| 5.4.2  | Mục tiêu 2 .....                               | 16 |
| 5.5  | Nhận xét .....                                 | 18 |
| TỔNG KẾT .....                                   |  | 21 |
| 1.   | Kết luận .....                                 | 21 |
| 1.1  | Về hệ thống .....                              | 21 |
| 1.2  | Về module machine learning .....               | 21 |
| 2.   | Kiến nghị .....                                | 21 |

# CHƯƠNG 1: TỔNG QUAN DỰ ÁN

## 1.1 Định nghĩa dự án

Hệ thống chăm sóc sức khỏe cho bệnh nhân tiểu đường là nơi để kết nối giữa bệnh nhân và các bác sĩ để giúp cho quá trình quản lý bệnh tiểu đường phù hợp về trình độ, giá cả và hiệu quả.

## 1.2 Mô tả vấn đề

Trong cuộc sống hiện đại ngày nay, sự gia tăng các bệnh ảnh hưởng tới sức khỏe lẫn công việc, đặt biệt trong đó là bệnh tiểu đường. Chúng xuất hiện từ các độ tuổi cũng như giới tính và hơn hết tỉ lệ mắc bệnh ngày càng tăng do chủ quan trong lối sinh hoạt hằng ngày. Để khi những biểu hiện của bệnh xuất hiện thì lúc đó chúng ta mới nhận thì đã không kịp, hay là chủ quan với bệnh dẫn tới những biến chứng nguy hiểm về tim, mắt, thận hay xương khớp.

Để giải quyết vấn đề trên việc xây dựng một hệ thống sẽ giúp cho những người chưa và đang mắc bệnh tiểu đường giúp quản lý bệnh trong khâu thức ăn, thuốc, vận động nhằm làm chậm hoặc ngăn ngừa các biến chứng nguy hiểm và thay đổi lối sống lành mạnh hơn.

## 1.3 Đề xuất giải pháp

### 1.3.1 Mục tiêu dự án

Phân tích, thiết kế và xây dựng một hệ thống chăm sóc sức khỏe cho bệnh nhân tiểu đường và áp dụng machine learning vào.

### 1.3.2 Phạm vi dự án

Giao tiếp giữa người dùng và hệ thống thông qua một ứng dụng trên nền tảng desktop, hệ thống có thể áp dụng với nhu cầu cá nhân. Hệ thống xoay quanh việc hỗ trợ người dùng kiểm soát cũng như nâng cao ý thức về bệnh tiểu đường.

## CHƯƠNG 2: PHÂN LOẠI VÀ PHÂN ĐOẠN HÌNH ẢNH

### 2.1 Phân loại hình ảnh

#### 2.1.1 Khái niệm

Phân loại giữa các đối tượng là một nhiệm vụ khá dễ dàng đối với chúng tôi, nhưng nó đã được chứng minh là một nhiệm vụ phức tạp cho máy móc và do đó phân loại hình ảnh đã là một nhiệm vụ quan trọng trong lĩnh vực thị giác máy tính. Phân loại hình ảnh đề cập đến việc dán nhãn hình ảnh thành một trong một số lớp được xác định trước.

Có khả năng *không có* số lớp trong đó một hình ảnh nhất định có thể được phân loại. Kiểm tra và phân loại hình ảnh theo cách thủ công có thể là một nhiệm vụ tẻ nhạt đặc biệt là khi chúng có số lượng lớn (ví dụ 10.000) và do đó sẽ rất hữu ích nếu chúng ta có thể tự động hóa toàn bộ quá trình này bằng thị giác máy tính.

**Một số ví dụ về phân loại hình ảnh bao gồm:**

- Dán nhãn X-quang là ung thư hay không (phân loại nhị phân).
- Phân loại chữ số viết tay (phân loại nhiều lớp).
- Gán tên cho một bức ảnh của một khuôn mặt (phân loại nhiều lớp).

#### 2.1.2 Kỹ thuật phân loại ảnh

Có nhiều thuật toán khác nhau được ứng dụng trong việc phân loại hình ảnh. Các thuật toán này được chia thành hai nhóm chính là Học có giám sát (supervised learning) và Học không giám sát (unsupervised learning).

##### *Phân loại có giám sát*

Trong học máy có giám sát, thuật toán được huấn luyện trên một tập hình ảnh đã được dán nhãn. Từ dữ liệu mẫu này, thuật toán có thể trích xuất thông tin, phục vụ phân loại ngay cả những hình ảnh chưa từng nhìn thấy trước đó.

Xuyên suốt quá trình đào tạo, đặc điểm của ma trận hình ảnh sẽ được trích xuất dưới dạng dữ liệu quan trọng để đưa vào xử lý. Các đặc điểm này đại diện cho hình ảnh trong không gian chiều thấp (lower-dimensional feature space) và là cơ sở để thuật toán tiến hành phân loại.

Trong quá trình đánh giá, các đặc điểm của ảnh thử nghiệm được thu thập và tái phân loại với sự hỗ trợ của mạng thần kinh nhân tạo. Hệ thống lúc này đã có thể nhận biết các đặc điểm điển hình của mọi lớp hình ảnh mà nó được đào tạo.

Các phương pháp phân loại phổ biến dựa trên học có giám sát bao gồm:

- Support Vector Machines
- Decision Trees
- K Nearest Neighbors

Các mạng nơ-ron thường được sử dụng để phân loại hình ảnh có giám sát bao gồm AlexNet, ResNet, DenseNet và Inception.

Đối với phân loại có giám sát, việc dán nhãn dữ liệu đóng vai trò quan trọng. Độ chính xác của dữ liệu được dán nhãn quyết định phần lớn hiệu suất của mô hình học máy. Các thuật toán phân loại có giám sát có thể được chia thành hai mục nhỏ hơn dựa trên nhãn dữ liệu.

### ***Phân loại nhãn đơn***

Phân loại nhãn đơn (Single-label classification) là tác vụ phổ biến nhất trong phân loại ảnh có giám sát. Theo đó, mỗi hình ảnh được đại diện bởi một nhãn/chú thích (a single label or annotation). Mô hình xuất ra một giá trị hoặc dự đoán duy nhất cho mỗi hình ảnh mà nó xử lý.

Đầu ra từ mô hình là mã hóa One-hot (từng giá trị được biến đổi thành các đặc trưng nhị phân chỉ chứa giá trị 1 hoặc 0). Mã hóa One-hot có độ dài bằng số lớp và giá trị biểu thị xác suất hình ảnh thuộc về lớp này.

Hàm Softmax được sử dụng để đảm bảo các xác suất tổng bằng một và xác suất tối đa được chọn làm đầu ra của mô hình. Mặc dù Softmax không có giá trị về mặt dự đoán, nhưng nó giúp ràng buộc đầu ra giữa 1 và 0, nhờ vậy, có thể đánh giá độ tin cậy của mô hình từ điểm Softmax.

Một số ví dụ về bộ dữ liệu phân loại nhãn đơn bao gồm MNIST, SVHN, ImageNet, v.v.

Phân loại nhãn đơn có thể được xếp vào phân loại đa lớp (Multiclass classification) hoặc phân loại nhị phân (binary classification).

### ***Phân loại đa nhãn***

Phân loại đa nhãn là một tác vụ phân loại trong đó mỗi hình ảnh có thể chứa nhiều hơn một nhãn hoặc một số hình ảnh chứa đồng thời tất cả các nhãn. Phân loại đa nhãn xuất hiện phổ biến trong lĩnh vực xử lý hình ảnh y tế, khi một bệnh nhân có thể được chẩn đoán mắc nhiều bệnh dựa trên dữ liệu chụp X-quang.

### ***Phân loại không giám sát***

Trong học máy không giám sát, thuật toán chỉ sử dụng dữ liệu thô để đào tạo. Các nhãn phân loại thường không xuất hiện trong kiểu học này và mô hình học bằng cách nhận dạng các mẫu trong tập dữ liệu huấn luyện.

Giống như phân loại có giám sát, các phương pháp dựa trên không giám sát cũng liên quan đến bước trích xuất đặc điểm với các thông tin chi tiết nhất về hình ảnh. Các đặc điểm này sau đó được xử lý bằng các phương pháp phân cụm tham số (Gaussian Mixture Models) và phi tham số (K-means) hoặc các thuật toán học không giám sát khác.

Các thuật toán và kỹ thuật phân loại của thị giác máy tính không chỉ giới hạn trong dữ liệu hình ảnh 2D đơn giản mà còn mở rộng ra dưới dạng Video và ảnh 3D.

### ***Phân loại video***

Khác với Phân loại hình ảnh, vốn chỉ sử dụng các thuật toán Xử lý ảnh và Mạng thần kinh tích chập (CNN), các tác vụ Phân loại video sử dụng cả dữ liệu hình ảnh và dữ liệu tạm thời (liên quan đến thời gian).

Để có thể áp dụng trực tiếp thuật toán phân loại hình ảnh tiêu chuẩn, mô hình phân loại video sẽ khai thác mối quan hệ giữa các khung hình khác nhau. Theo đó, các mạng thần kinh phù hợp với dữ liệu chuỗi thời gian như LSTM (Bộ nhớ ngắn hạn dài)

và RNN (Mạng thần kinh hồi quy) sẽ kết hợp với CNN để phân tích mối quan hệ thời gian giữa các khung.

### ***Phân loại 3D***

Khác biệt cơ bản của phân loại 3D khi so với phân loại 2D nằm ở cấu trúc của CNN và bản chất chuyển động của hạt nhân trượt (sliding kernel).

Nhân trong phân loại dữ liệu 3D cũng là 3D và di chuyển dọc theo cả ba trục, khác với chuyển động thẳng hai trục trong CNNs 2D. CNN có khả năng nắm bắt dữ liệu không gian rất tốt, và do đó dễ dàng xử lý khi dữ liệu được đặt cách nhau trên ba trục.

Bộ dữ liệu phân loại 3D dễ dàng được tìm thấy trong lĩnh vực y tế (ví dụ: ảnh cộng hưởng từ não) và dữ liệu cấu trúc của các đại phân tử thu được từ Kính hiển vi điện tử lạnh.

### **2.1.3 Cấu trúc của bài toán phân loại hình ảnh**

#### **1. Xử lý trước hình ảnh**

Mục đích của quá trình này là cải thiện dữ liệu hình ảnh (tính năng) bằng cách ngăn chặn các biến dạng không mong muốn và tăng cường một số tính năng hình ảnh quan trọng để các mô hình Thị giác máy tính của chúng tôi có thể hưởng lợi từ dữ liệu được cải thiện này để hoạt động.

#### **2. Phát hiện một đối tượng**

Phát hiện đề cập đến việc nội địa hóa một đối tượng có nghĩa là phân đoạn hình ảnh và xác định vị trí của đối tượng quan tâm.

#### **3. Trích xuất tính năng và đào tạo**

Đây là một bước quan trọng trong đó các phương pháp thống kê hoặc học sâu được sử dụng để xác định các mẫu thú vị nhất của hình ảnh, các tính năng có thể là duy nhất cho một lớp cụ thể và sau này sẽ giúp mô hình phân biệt giữa các lớp khác nhau. Quá trình này, nơi mô hình tìm hiểu các tính năng từ bộ dữ liệu được gọi là đào tạo mô hình.

#### **4. Phân loại đối tượng**



Bước này phân loại các đối tượng được phát hiện thành các lớp được xác định trước bằng cách sử dụng một kỹ thuật phân loại phù hợp để so sánh các mẫu hình ảnh với các mẫu mục tiêu.

## 2.2 Phân đoạn hình ảnh

### 2.2.1 Khái niệm

Phân vùng ảnh (Image segmentation) là một phương pháp mà trong đó, hình ảnh kỹ thuật số được chia thành nhiều nhóm con khác nhau được gọi là segments. Mục tiêu của phân vùng ảnh là làm giảm độ phức tạp của hình ảnh, giúp cho quá trình xử lý hoặc phân tích hình ảnh sau đó trở nên đơn giản hơn. Nói một cách dễ hiểu, phân vùng là dán nhãn cho từng pixel.

Tất cả các yếu tố hình ảnh hoặc pixel thuộc cùng một danh mục sẽ có chung một nhãn. Ví dụ: Đối với bài toán phát hiện đối tượng, thay vì xử lý toàn bộ hình ảnh, máy có thể chỉ thực hiện trên một đoạn được chọn bởi thuật toán phân vùng. Điều này sẽ ngăn máy xử lý toàn bộ hình ảnh, do đó làm giảm thời gian suy luận.



Hình 1: Quá trình xử lý ảnh

### 2.2.2 Cách tiếp cận

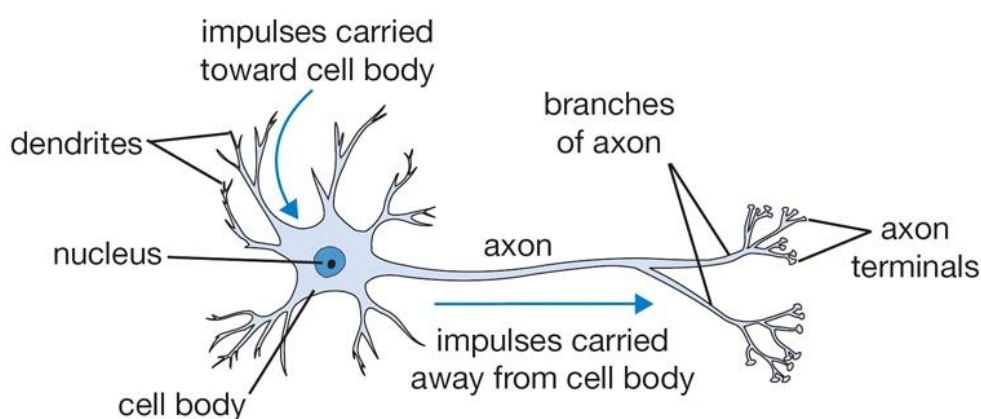
- Cách tiếp cận tương đồng (Similarity approach), có nghĩa là phát hiện sự tương đồng giữa các pixel hình ảnh để tạo thành một phân đoạn, dựa trên một ngưỡng. Các thuật toán học máy như phân cụm thường dựa trên kiểu tiếp cận này để phân vùng một hình ảnh.
- Cách tiếp cận gián đoạn (Discontinuity approach): Cách tiếp cận này dựa trên sự gián đoạn của các giá trị cường độ pixel trong hình ảnh. Các kỹ thuật phát hiện đường, điểm và cạnh sử dụng kiểu tiếp cận gián đoạn để thu được các kết quả phân vùng trung gian. Kết quả này sau đó có thể được xử lý để cho ra hình ảnh được phân vùng cuối cùng.

## CHƯƠNG 3: MẠNG TÍCH CHẬP

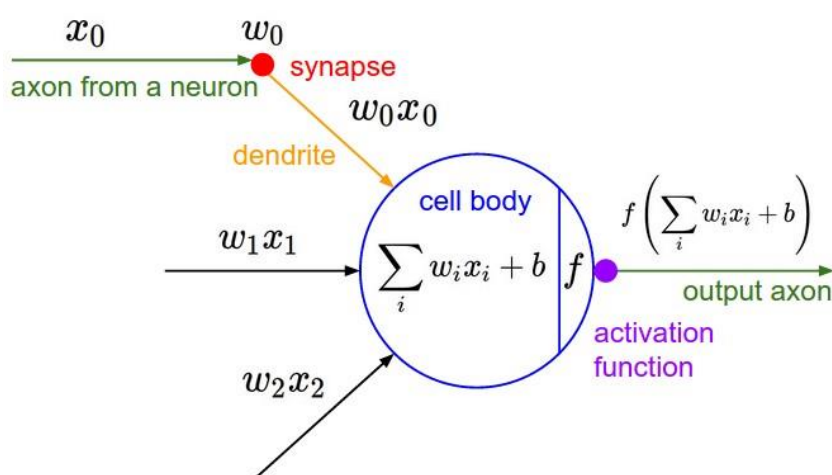
### 3.1 Mạng thần kinh nhân tạo

Lấy cảm hứng từ các tính chất của mạng thần kinh sinh học, Mạng thần kinh nhân tạo là thuật toán học tập thống kê và được sử dụng cho nhiều nhiệm vụ khác nhau, từ các nhiệm vụ phân loại tương đối đơn giản đến thị giác máy tính và nhận dạng giọng nói. ANN được thực hiện như một hệ thống các yếu tố xử lý liên kết với nhau, được gọi là các nút, có chức năng tương tự như các tế bào thần kinh sinh học.

Các kết nối giữa các nút khác nhau có các giá trị số, được gọi là trọng số và bằng cách thay đổi các giá trị này một cách có hệ thống, mạng cuối cùng có thể xấp xỉ hàm mong muốn.



Hình 3: Mạng thần kinh sinh học



Hình 4: Công thức toán học của nút trong mạng ANN

Các lớp ẩn có thể được coi là máy dò tính năng riêng lẻ, nhận ra ngày càng nhiều mẫu phức tạp hơn trong dữ liệu vì nó được lan truyền khắp mạng.

Ví dụ, nếu mạng được giao nhiệm vụ nhận dạng khuôn mặt, lớp ẩn đầu tiên có thể hoạt động như một máy dò đường, lớp ẩn thứ hai lấy các đường này làm đầu vào và đặt chúng lại với nhau để tạo thành mũi, lớp ẩn thứ ba lấy mũi và khớp với nó bằng mắt và vân vân, cho đến khi cuối cùng toàn bộ khuôn mặt được xây dựng. Hệ thống phân cấp này cho phép mạng cuối cùng nhận ra các đối tượng rất phức tạp.

Mạng noron nhân tạo (Artificial Neural Networks) mô phỏng lại mạng noron sinh học là một cấu trúc khối gồm các đơn vị tính toán đơn giản được liên kết chặt chẽ với nhau trong đó các liên kết giữa các noron quyết định chức năng của mạng.

### **3.2 Mạng tích chập**

Mạng thần kinh phức tạp (CNN) là một kiến trúc đặc biệt của mạng lưới thần kinh nhân tạo. CNNs sử dụng một số tính năng của vỏ não thị giác và do đó đã đạt được kết quả hiện đại trong các nhiệm vụ thị giác máy tính. Chúng ta hãy đề cập đến việc sử dụng CNN chi tiết hơn. Mạng lưới thần kinh phức tạp bao gồm hai yếu tố rất đơn giản, cụ thể là các lớp phức tạp và các lớp gộp lại.

Mặc dù đơn giản, có những cách gần như vô hạn để sắp xếp các lớp này cho một vấn đề thị giác máy tính nhất định. Các yếu tố của một mạng lưới thần kinh phức tạp, chẳng hạn như các lớp phức tạp và gộp lại, tương đối đơn giản để hiểu.

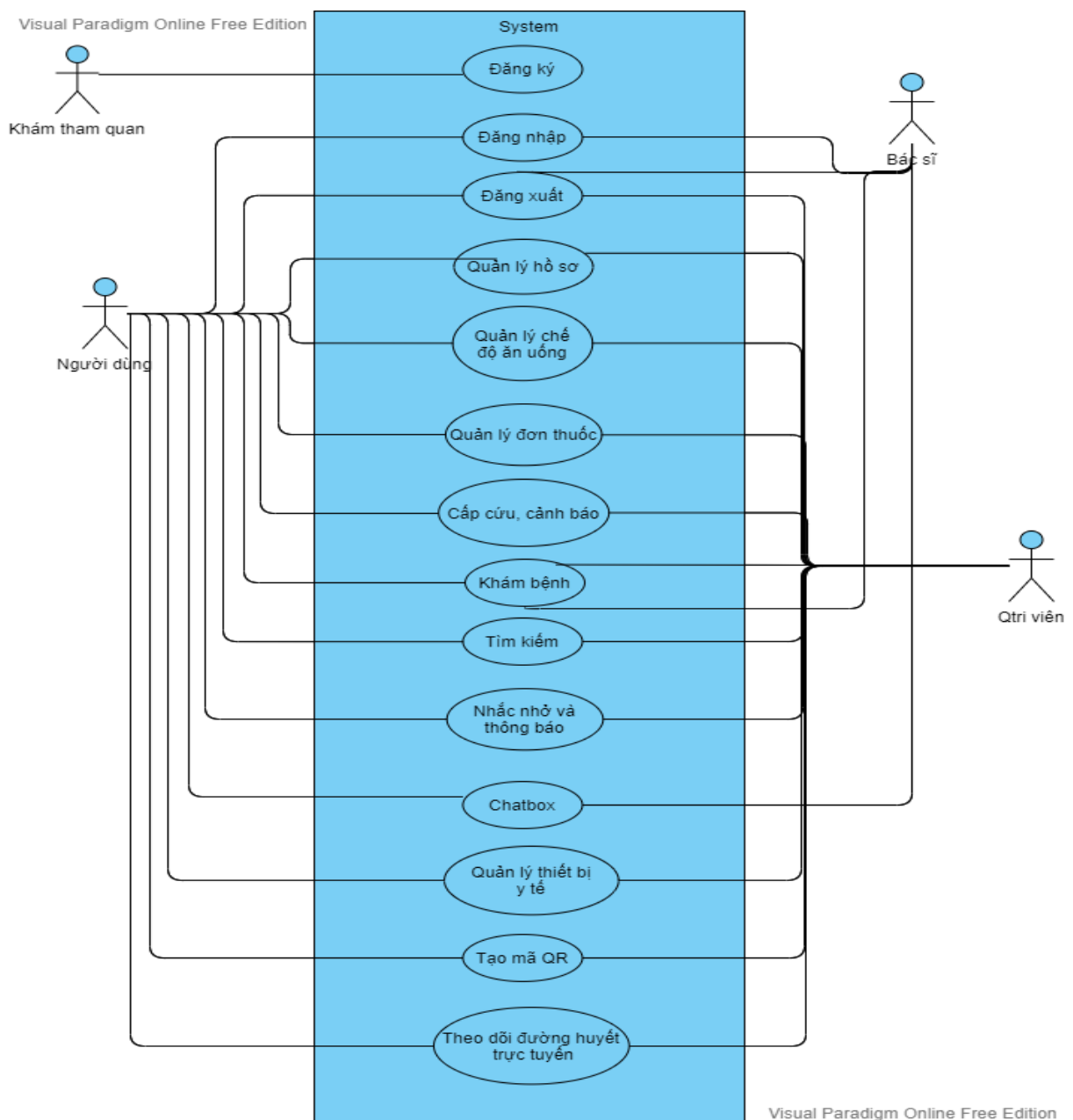
Phản thách thức của việc sử dụng các mạng thần kinh phức tạp trong thực tế là làm thế nào để thiết kế các kiến trúc mô hình sử dụng tốt nhất các yếu tố đơn giản này.

## CHƯƠNG 4: XÂY DỰNG HỆ THỐNG

### 4.1 Đề xuất kỹ thuật

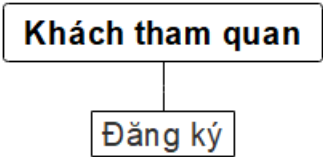
- \* Ngôn ngữ lập trình: Python
- \* Kiến trúc phần mềm: MVC
- \* Develop tool: PyCharm
- \* Manage source code tools: GitHub

### 4.2 Usecase hệ thống

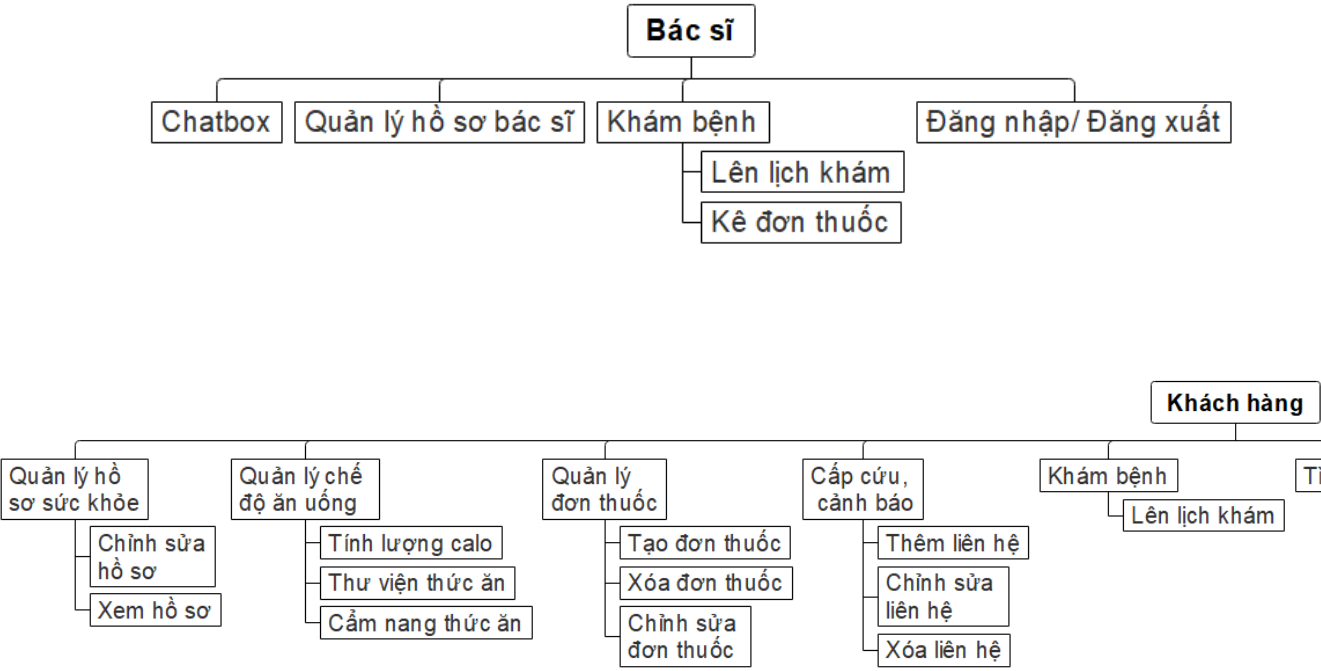


Hình 7: Usecase hệ thống

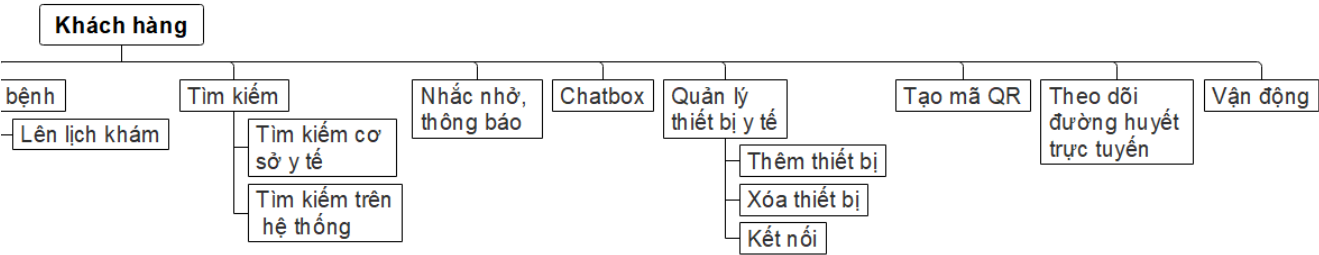
### 4.3 Product backlog



Hình 8: Mô-đun khách truy cập



Hình 9: Mô-đun bác sĩ



Hình 10: Mô-đun khách hàng

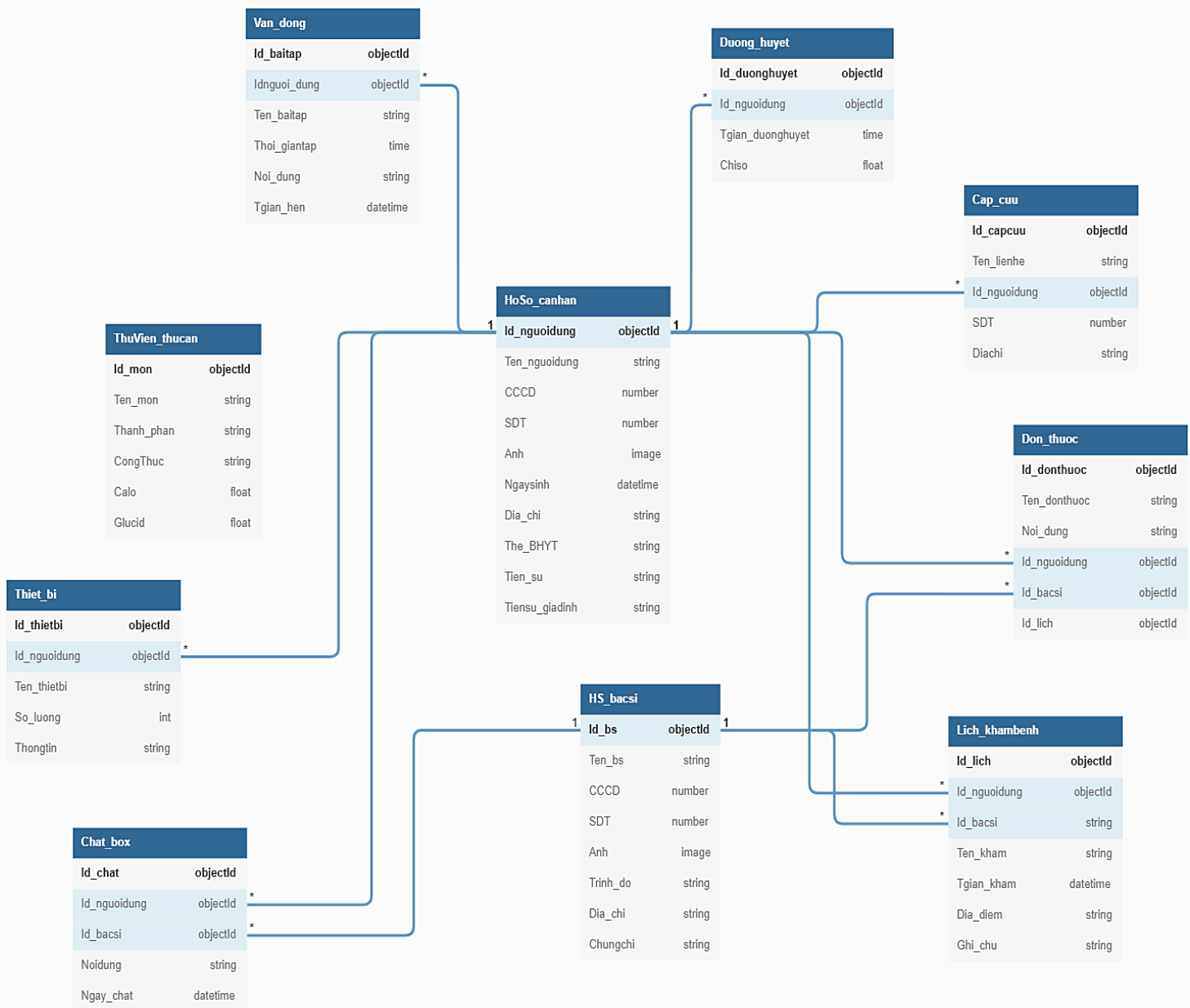
#### 4.4 Cơ sở dữ liệu

##### 4.4.1 Tổng quan về bảng

| No | Tên bảng         | Mô tả  |
|----|------------------|--|
| 1  | Hồ sơ cá nhân    | Thông tin sơ yếu về người dùng                   |
| 2  | Thư viện thức ăn | Thông tin thức ăn cho bệnh nhân tiểu đường       |
| 3  | Vận động         | Thông tin các bài tập vận động                   |
| 4  | Đơn thuốc        | Thông tin về thuốc uống                          |
| 5  | Thiết bị         | Thông tin các thiết bị của người dùng            |
| 6  | Hồ sơ bác sĩ     | Thông tin về bác sĩ                              |
| 7  | Lịch khám bệnh   | Thông tin về lịch khám bệnh                      |
| 8  | Cấp cứu          | Thông tin liên hệ khi có cảnh báo về sức khỏe    |
| 9  | Chatbox          | Nội dung giao tiếp giữa bác sĩ và người dùng     |
| 10 | Đường huyết      | Chỉ số đường huyết của người dùng từng thời điểm |

*Bảng 1: Tổng quan về bảng*

## 4.4.2 Sơ đồ mối quan hệ bảng



Hình 11: Sơ đồ mối quan hệ bảng

## CHƯƠNG 5: XÂY DỰNG MODULE MACHINE LEARNING

### 5.1 Vấn đề

Để tính toán lượng calo có trong thực phẩm trái cây, chúng ta cần đo khối lượng của chúng và thực hiện phép tính để tính được lượng calo. Quá trình này chúng ta phải thực hiện nhiều lần tính toán cũng như cân đo, việc này làm tốn khá nhiều thời gian.

### 5.2 Phương pháp giải quyết

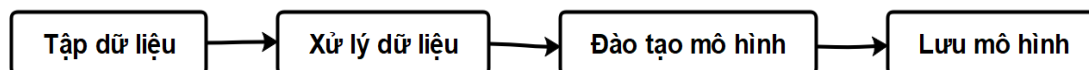
Để giải quyết vấn đề trên một cách đơn giản bằng cách chúng ta chụp một bức ảnh chứa một thực phẩm với ngón cái, ở đây ngón cái được coi là thước đo tiêu chuẩn để đo được kích thước thực phẩm, áp dụng machine learning vào nhận ra thực phẩm đó để suy ra lượng mật độ dinh dưỡng và ước tính khối lượng, lượng calo.

### 5.3 Mục tiêu

- Mục tiêu 1: Đào tạo model nhận dạng các loại thực phẩm (gồm 7 loại thực phẩm: táo, chuối, cà rốt, dưa chuột, hành, cam, cà chua).
- Mục tiêu 2: Ước tính trọng lượng và lượng calo của thực phẩm.

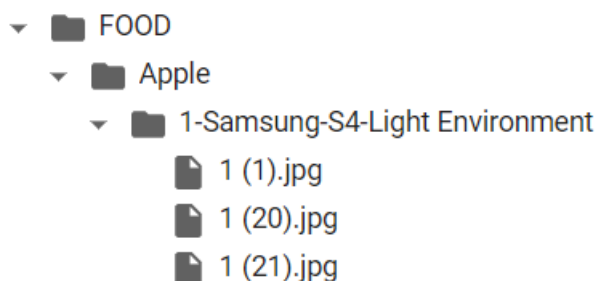
### 5.4 Mô hình thực nghiệm và triển khai

#### 5.4.1 Mục tiêu 1



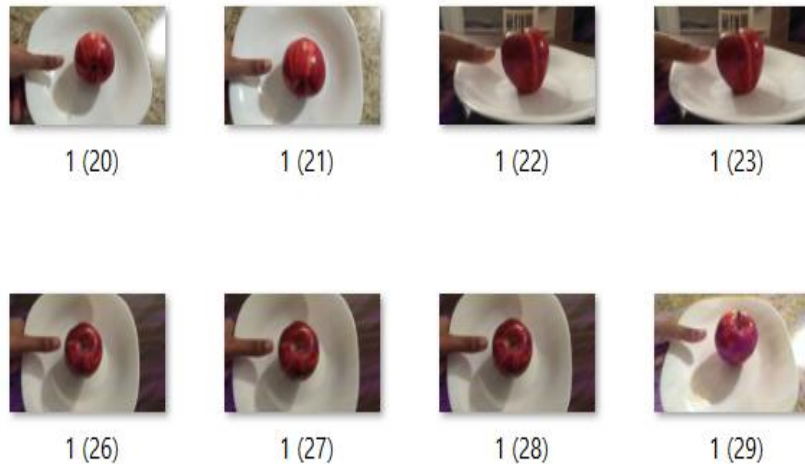
##### 5.4.1.1 Tập dữ liệu

Tập dữ liệu có cấu trúc như sau:



Bộ dữ liệu chứa 7 loại thực phẩm, trong mỗi thư mục thực phẩm chứa các thư mục tên thiết bị chụp ảnh và trong đó là ảnh về thực phẩm. Mỗi tấm ảnh chứa thực phẩm và ngón tay cái.





#### 5.4.1.2 Xử lý dữ liệu

- Input: Đường dẫn tới thư mục chứa file hình ảnh của tập dữ liệu
- Quá trình: Tạo vòng lặp để đọc ảnh và label của chúng, sau đó resize ảnh về kích thước 400x400, tiếp đến chuyển chúng qua mảng và lưu kèm với label của bức ảnh.
- Output: Bộ dữ liệu được trả về mảng chứa dữ liệu ảnh và label

```
def create_train_data(path):
    training_data = []
    folders=os.listdir(path)[0:no_of_fruits]
    for i in range(len(folders)):
        label = [0 for i in range(no_of_fruits)]
        label[i] = 1
        print(folders[i])
        k=0
        for j in glob.glob(path+"/"+folders[i]+"/*/*.jpg"):
            if(k==no_of_images):
                break
            k=k+1
            img = cv2.imread(j)
            img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))
            print(img,label)
            training_data.append([np.array(img),np.array(label)])
    return training_data,folders
```

#### 5.4.1.3 Đào tạo mô hình và lưu mô hình

- Input: Mảng chứa dữ liệu và label tương ứng
- Quá trình: Chia tập dữ liệu thành 2 phần train và set với tỉ lệ 80:20. Sau đó đưa vào một model mạng CNN để trải qua quá trình huấn luyện

```

size=int(len(training_data)*percentage)
print(size)
train = training_data[:-size]
test=training_data[-size:]
X = np.array([i[0] for i in train]).reshape(-1,IMG_SIZE,IMG_SIZE,3)
Y = [i[1] for i in train]
test_x = np.array([i[0] for i in test]).reshape(-1,IMG_SIZE,IMG_SIZE,3)
test_y = [i[1] for i in test]

```

```

model=get_model(IMG_SIZE,no_of_fruits,LR)

model.fit({'input': X}, {'targets': Y}, n_epoch=10, validation_set=({'input': test_x},
                                                                    {'targets': test_y}),
        snapshot_step=500, show_metric=True, run_id=MODEL_NAME)

model_save_at=os.path.join("model",MODEL_NAME)
model.save(model_save_at)

```

Trong đó kiến trúc CNN là :

```

convnet = input_data(shape=[None, IMG_SIZE, IMG_SIZE, 3], name='input')

convnet = conv_2d(convnet, 32, 5, activation='relu')

convnet = max_pool_2d(convnet, 5)

convnet = conv_2d(convnet, 64, 5, activation='relu')

convnet = max_pool_2d(convnet, 5)

convnet = conv_2d(convnet, 128, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)

convnet = conv_2d(convnet, 64, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)

convnet = conv_2d(convnet, 32, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)

convnet = fully_connected(convnet, 1024, activation='relu')
convnet = dropout(convnet, 0.8)

convnet = fully_connected(convnet, no_of_fruits, activation='softmax')
convnet = regression(convnet, optimizer='adam', learning_rate=LR, loss='categorical_crossentropy', name='targets')

model = tflearn.DNN(convnet, tensorboard_dir='log')

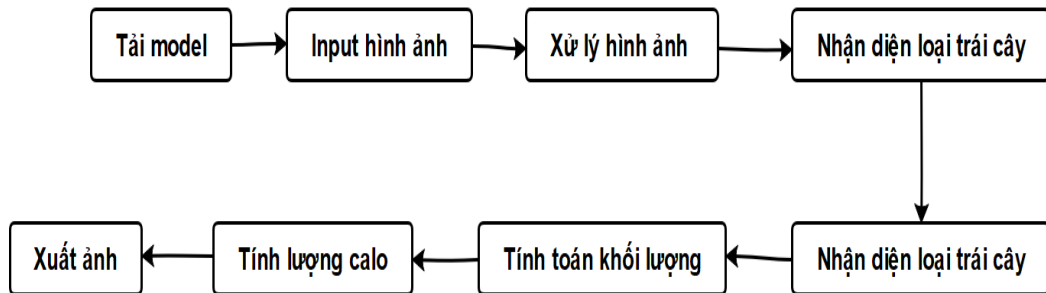
```

- Output: Mô hình được lưu dưới tên *Fruits\_detector-0.001-5conv-basic.model*

```
print("Model Save At",model_save_at)
```

Model Save At model/Fruits\_detector-0.001-5conv-basic.model

## 5.4.2 Mục tiêu 2



### 5.4.2.1 Tải model

- Input: Tên thư mục chứa model nhận dạng thực phẩm đã training.
- Quá trình: Tải model mạng CNN và model đã train vào.
- Code thực hiện:

```
MODEL_NAME = 'Fruits_dectector-{}-{}.model'.format(LR, '5conv-basic')  
  
model_save_at=os.path.join("model",MODEL_NAME) # input model  
  
model=get_model(IMG_SIZE,no_of_fruits,LR) # input model cnn  
  
model.load(model_save_at) # load model  
labels=list(np.load('labels.npy')) #input lable
```

### 5.4.2.2 Input và xử lý hình ảnh

- Input: Đường link ảnh
- Quá trình: Đọc và resize ảnh về kích thước 400\*400.
- Code thực hiện:

```
test_data='test_image.JPG'  
img=cv2.imread(test_data)  
img1=cv2.resize(img,(IMG_SIZE,IMG_SIZE))
```

### 5.4.2.3 Nhận diện loại trái cây

- Input: Bức ảnh đã được resize 400\*400.
- Quá trình: Tiến hành nhận diện thực phẩm

- Output: Tên thực phẩm

- Code thực hiện:

```
model_out=model.predict([img1]) # nhận diện trái cây
result=np.argmax(model_out) #
name=labels[result] # trả về tên trái cây
```

#### 5.4.2.4 Tính toán khối lượng

- Input: Label trái cây, các thông số khác

- Quá trình: Đối với các thực phẩm hình cầu chúng ta áp dụng công thức để tính ra bán kính và từ đó tính ra thể tích, đối với hình trụ chúng ta tính ra chiều cao và bán kính.

- Output: xuất ra giá trị volume

- Code thực hiện:

```
def getVolume(label, area, skin_area, pix_to_cm_multiplier, fruit_contour):
    area_fruit = (area/skin_area)*skin_multiplier
    label = int(label)
    volume = 100
    if label == 1 or label == 5 or label == 7 or label == 6 :
        radius = np.sqrt(area_fruit/np.pi)
        volume = (4/3)*np.pi*radius*radius*radius

    if label == 2 or label == 4 or (label == 3 and area_fruit > 30):
        fruit_rect = cv2.minAreaRect(fruit_contour)
        height = max(fruit_rect[1])*pix_to_cm_multiplier
        radius = area_fruit/(2.0*height)
        volume = np.pi*radius*radius*height

    if (label==4 and area_fruit < 30) :
        volume = area_fruit*0.5

    return volume
```

#### 5.4.2.5 Tính toán lượng calo

- Input: Tên label thực phẩm và kích thước.

- Quá trình: Dựa vào mật độ dinh dưỡng cũng như lượng calo có trong thực phẩm và áp dụng công thức tính lượng calo theo khối lượng

- Output: Lượng calo dự đoán

- Code thực hiện :

```
def getCalorie(label, volume):
    calorie = calorie_dict[int(label)]
    density = density_dict[int(label)]
    mass = volume*density*1.0
    calorie_tot = (calorie/100.0)*mass
    return calorie_tot
```

#### 5.4.2.6 Xuất ảnh

- Input: Đầu vào là ảnh và lượng calo đã tính
- Quá trình: Dùng thư viện matplotlib để show ảnh
- Output: Ảnh bao gồm tên thực phẩm và lượng calo dự đoán

Banana(183.23kcal)



- Code thực hiện:

```
import matplotlib.pyplot as plt
plt.imshow(img)
plt.title('{}({}kcal)'.format(name,cal))
plt.axis('off')
plt.show()
```

## 5.5 Nhận xét

### 5.5.1 Kết quả từ ước tính lượng calo

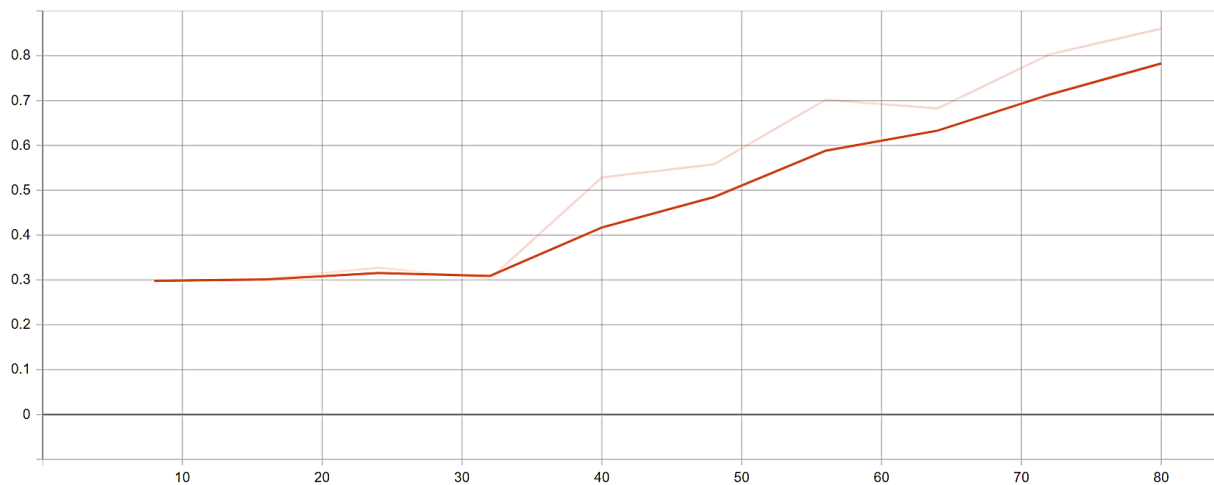
Dưới đây là bảng kết quả so sánh giữa lượng calo thực thể và lượng calo ước tính từ hình ảnh.

| Thực phẩm | Calo  | Lượng calo ước tính |
|-----------|-------|---------------------|
| Táo       | 53.96 | 40.42               |

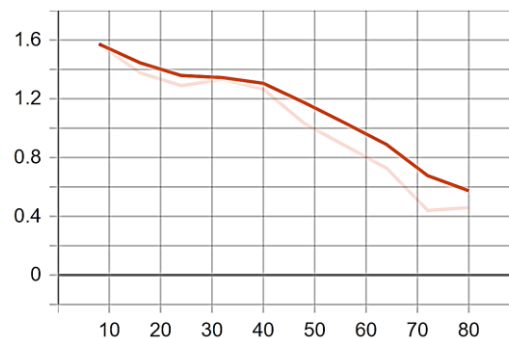
|           |        |        |
|-----------|--------|--------|
| Chuối     | 170.88 | 188.81 |
| Cà rốt    | 31.16  | 26.28  |
| Dưa chuột | 29.44  | 37.65  |
| Hành      | 44.88  | 37.13  |
| Cam       | 69.09  | 71.92  |
| Cà chua   | 17.46  | 13.82  |

### 5.5.2 Kết quả cho quá trình đào tạo model

Sau đây là độ chính xác cũng như tổn thất của mô hình khi đào tạo.



*Độ chính xác*



*Sự tổn thất*

### 5.5.3 Hạn chế:

- Trọng lượng thực tế và lượng calo không thể tìm thấy do chất lượng hình ảnh
- Khó tìm góc thích hợp giữa trái cây và máy ảnh
- Điều kiện ánh sáng tức là thay đổi pixel liên quan đến ánh sáng

### 5.5.4 Phạm vi:

- Ước tính lượng calo từ tất cả các loại trái cây.
- Giảm thiểu sai số ước tính lượng calo



# TỔNG KẾT

## 1. Kết luận

### 1.1 Về hệ thống

Về hệ thống được xây dựng theo mô hình mvc và việc giao tiếp giữa người dùng thông qua ứng dụng desktop còn nhiều mặt hạn chế về độ linh hoạt cũng như sự thuận tiện trong việc sử dụng. Hệ thống chỉ áp dụng một module machine learning vào nên cần áp dụng nhiều hơn và việc tạo một lịch trình hay một sơ đồ điều trị bệnh mà hệ thống chưa làm được là điều cần thiết.

### 1.2 Về module machine learning

Về việc nhận dạng thực phẩm chỉ mới dừng lại 7 loại, cần tăng thêm số lượng đồng thời có thể áp dụng một số mạng nơ ron để đồng thời vừa nhận dạng vừa tính toán được khối lượng. Quá trình đào tạo model tỉ lệ đúng còn nhiều thiếu sót và việc là đa dạng hóa lượng dữ liệu đầu vào giúp cho tỉ lệ nhận diện thực phẩm được nâng cao.

## 2. Kiến nghị

Khi hệ thống đạt được hiệu quả trong việc quản lý tốt bệnh tiểu đường thì chúng ta có thể mở rộng mô hình cũng như đối tượng người sử dụng. Đồng thời sự phát triển về người dùng tăng dẫn đến lượng dữ liệu tăng lên, chúng ta có thể áp dụng các kiến thức về Big Data vào để xử lý lượng công việc nhanh chóng, ngoài ra không chỉ dừng lại việc chăm sóc một bệnh cụ thể thì chúng ta có thể nâng lên việc chăm sóc sức khỏe toàn diện. Đồng thời có thể áp dụng machine learning vào việc chăm sóc sức khỏe như: theo dõi đường huyết trực tuyến, dự đoán giá trị đường huyết, chẩn đoán bệnh võng mạc – biến chứng bệnh tiểu đường, ... để khai thác nhằm nâng cao chất lượng hệ thống.