TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

KHOA ĐIỆN TỬ

A circular logo with a star and a blue star with a red and yellow circle

Description automatically generated with medium confidence

**BÀI TẬP LỚN**

**LẬP TRÌNH PYTHON**

**BỘ MÔN: CÔNG NGHỆ THÔNG TIN**

**HỆ : ĐẠI HỌC CHÍNH QUY**

ĐỀ TÀI: SỬ DỤNG PYGAME XÂY DỰNG MỘT TỰA GAME PLATFORMER ĐƠN GIẢN

THÁI NGUYÊN - 2024

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

KHOA ĐIỆN TỬ



**BÀI TẬP LỚN LẬP TRÌNH PYTHON**

**BỘ MÔN : CÔNG NGHỆ THÔNG TIN**

ĐỀ TÀI: SỬ DỤNG PYGAME XÂY DỰNG MỘT TỰA GAME PLATFORMER ĐƠN GIẢN

|  |  |  |
| --- | --- | --- |
| **Giảng viên hướng dẫn** | **:** | **TS. Nguyễn Tuấn Linh** |
| **Họ và tên sinh viên** | **:** | **La Đức Thắng** |
| **Ngành học** | **:** | **Kỹ thuật Máy tính** |
| **MSSV** | **:** | **K215480106120** |
| **Lớp** | **:** | **K57KMT.01** |

THÁI NGUYÊN - 2024

|  |  |
| --- | --- |
| TRƯỜNG ĐHKTCN | CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM |
| KHOA ĐIỆN TỬ | Độc lập - Tự do - Hạnh phúc |

BÀI TẬP LỚN

LẬP TRÌNH PYTHON

**Sinh viên:** La Đức Thắng

**Lớp:** K57KMT **Khoá:** 2021-2025

**Bộ môn:** Công Nghệ Thông Tin

**Giáo viên hướng dẫn:** TS. Nguyễn Tuấn Linh

**Tên đề tài** : Xây dựng một tựa game platformer 2D đơn giản bằng Pygame.

Yêu cầu :

* Tạo nhân vật, vũ khí, kẻ thù
* Nhân vật có thể di chuyển, tấn công, có thanh HP
* Cuối mỗi chương có 1 màn boss

|  |  |  |
| --- | --- | --- |
| BCN KHOA | TRƯỞNG BỘ MÔN | GIÁO VIÊN HƯỚNG DẪN |
| (Ký và ghi rõ họ tên) | (Ký và ghi rõ họ tên) | (Ký và ghi rõ họ tên) |

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Xếp loại:……………… Điểm :……

Thái Nguyên, ngày….tháng…..năm 20....

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

MỤC LỤC

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN 4](#_Toc192258276)

[DANH MỤC TỪ VIẾT TẮT 7](#_Toc192258277)

[LỜI NÓI ĐẦU 8](#_Toc192258278)

[LỜI CẢM ƠN 9](#_Toc192258279)

[CHƯƠNG I: KHÁI QUÁT VỀ BÀI TOÁN 10](#_Toc192258280)

[1.1. Đặt vấn đề 10](#_Toc192258281)

[1.2. Hướng giải quyết 10](#_Toc192258282)

[CHƯƠNG II: PHÂN TÍCH BÀI TOÁN 11](#_Toc192258283)

[2.1. Phân tích bài toán 11](#_Toc192258284)

[2.2. Phương pháp học máy sử dụng 11](#_Toc192258285)

[2.2.1. Phương pháp DeepLearning 11](#_Toc192258286)

[2.2.2. VGGFace2 12](#_Toc192258287)

[2.2.3. InceptionResNetV1 14](#_Toc192258288)

[CHƯƠNG III: HUẤN LUYỆN MÔ HÌNH 17](#_Toc192258289)

[3.1. Một số khái niệm trong huấn luyện mô hình 17](#_Toc192258290)

[3.1.1. Batch 17](#_Toc192258291)

[3.1.2. Epoch 17](#_Toc192258292)

[3.1.3. Weights 18](#_Toc192258293)

[3.1.4. Iteration 18](#_Toc192258294)

[3.1.5. Loss 19](#_Toc192258295)

[3.1.6. Accuracy 19](#_Toc192258296)

[3.1.7. Overfitting và underfitting 20](#_Toc192258297)

[3.2. Huấn luyện mô hình 21](#_Toc192258298)

[3.2.1. Thu thập dữ liệu 21](#_Toc192258299)

[3.2.2. Tiền xử lý dữ liệu 21](#_Toc192258300)

[3.2.3. Xây dựng dataset 22](#_Toc192258301)

[3.2.4. Tiến hành huấn luyện 24](#_Toc192258302)

[3.3. Đánh giá mô hình 25](#_Toc192258303)

[CHƯƠNG IV: XÂY DỰNG CHƯƠNG TRÌNH 28](#_Toc192258304)

[4.1. Thiết kế hệ thống 28](#_Toc192258305)

[4.1.1. Phân tích chương trình 28](#_Toc192258306)

[4.1.2. Phân tích hệ thống 29](#_Toc192258307)

[4.1.3. Cơ sở dữ liệu 29](#_Toc192258308)

[4.1.4. Giao diện chương trình 30](#_Toc192258309)

[4.2. Kiểm thử chương trình 31](#_Toc192258310)

[CHƯƠNG V: TỔNG KẾT 34](#_Toc192258311)

[5.1. Kết quả đã đạt được 34](#_Toc192258312)

[5.2. Hướng phát triển của đề tài 34](#_Toc192258313)

[KẾT LUẬN 35](#_Toc192258314)

[TÀI LIỆU THAM KHẢO 36](#_Toc192258315)

# DANH MỤC TỪ VIẾT TẮT

1. CSDL: Cơ sở dữ liệu.
2. CNTT: Công nghệ thông tin

LỜI NÓI ĐẦU

Trong quá trình thực hiện đề tài, em đã cố gắng tìm hiểu, nghiên cứu và áp dụng những kiến thức, kỹ năng đã học để xây dựng một tựa game hoàn chỉnh, có khả năng hoạt động ổn định. Tuy nhiên, do kiến thức và kinh nghiệm còn hạn chế, chắc chắn sẽ không tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý, đánh giá từ quý thầy cô và các bạn để đề tài được hoàn thiện hơn.

LỜI CẢM ƠN

Trong suốt quá trình học tập và thực hiện bài tập lớn, em đã nhận được sự giúp đỡ tận tình của thầy giáo TS. Nguyễn Tuấn Linh trong bộ môn Tin học công nghiệp – Khoa Điện tử - Trường Đại học Kỹ thuật Công Nghiệp – Đại học Thái Nguyên. Em bày tỏ lòng biết ơn thầy đã tận tình giúp đỡ, hướng dẫn em trong thời gian thực hiện đề tài này.

Mặc dù đã cố gắng hết sức, song do điều kiện thời gian và kinh nghiệm thực tế còn ít, cho nên đề tài không thể tránh khỏi thiếu sót. Vì vậy, em rất mong nhận được sự đóng góp ý kiến của các thầy giáo, cô giáo và các bạn.

Em xin chân thành cảm ơn!

CHƯƠNG I: KHẢO SÁT BÀI TOÁN

## Khái niệm game platformer

Platformer là thể loại trò chơi mà nhân vật chính di chuyển trên mặt phẳng 2 chiều, thường có khả năng:

* Di chuyển trái/phải.
* Nhảy và chịu ảnh hưởng bởi trọng lực.
* Tương tác với các vật thể như platform, kẻ địch, vật phẩm.

Một số tựa game platformer nổi tiếng có thể kể đến như: Super Mario Bros, Sonic The Hedgehog, Hollow Knight, Blasphemous,…



Hình 1.1: Game Blasphemous

## Hướng giải quyết

CHƯƠNG II: PHÂN TÍCH BÀI TOÁN

## 2.1. Phân tích bài toán

Dữ liệu đầu vào:

* Bàn phím
* Sự kiện (Event Inputs)
* Tài nguyên game

Dữ liệu đầu ra:

* Hiển thị đồ họa (Nhân vật chính, kẻ địch, nền, platform, đạn, hiệu ứng, HUD animation khi di chuyển, nhảy, tấn công.)
* Thay đổi trạng thái game

## 2.2. Các thư viện sử dụng

2.2.1. Pygame

CHƯƠNG III: THIẾT KẾ CHƯƠNG TRÌNH

## 3.1. Một số khái niệm trong huấn luyện mô hình

### 3.1.1. Batch

Batch là một tập hợp con của dữ liệu huấn luyện, được đưa vào mô hình trong một vòng lặp huấn luyện (training iteration).

* Vì tập dữ liệu thường quá lớn, không thể xử lý hết một lần → Ta chia nhỏ dữ liệu thành nhiều batch để xử lý dần.
* Mỗi batch được đưa vào mô hình để tính toán và cập nhật trọng số.
* Sau khi tất cả các batch được xử lý, ta hoàn thành một epoch.

### 3.1.2. Epoch

Epoch là một vòng lặp qua toàn bộ tập dữ liệu huấn luyện khi đào tạo mô hình Machine Learning.

Trong quá trình huấn luyện mô hình, dữ liệu huấn luyện thường rất lớn, không thể đưa hết vào mô hình một lần. Khi đó dữ liệu sẽ được chia thành nhiều batch nhỏ (Mini-Batch). Một epoch hoàn tất khi toàn bộ dữ liệu đã được huấn luyện qua mô hình 1 lần (qua nhiều batch). Để mô hình học tốt hơn, nhiều epoch được lặp lại để mô hình tối ưu trọng số.

Khi dữ liệu đi qua 1 epoch, sẽ xảy ra các bước sau:

* Chia dữ liệu thành batch (nếu dùng mini-batch).
* Forward Propagation: Mô hình dự đoán đầu ra từ dữ liệu đầu vào.
* Tính toán lỗi (Loss Calculation): So sánh đầu ra dự đoán với giá trị thực tế để tính toán mức độ sai lệch.
* Backward Propagation: Tính gradient và cập nhật trọng số (weights) bằng thuật toán tối ưu (SGD, Adam, RMSprop, v.v.).

Lặp lại quá trình này cho tất cả các batch trong epoch.

### 3.1.3. Weights

Weights (trọng số) thường đề cập đến các giá trị số được sử dụng để điều chỉnh mức độ ảnh hưởng của một yếu tố nào đó trong mô hình.

Trong Deep Learning, weights là các tham số quan trọng nhất trong mạng nơ-ron nhân tạo (Artificial Neural Network - ANN). Chúng quyết định cách mà mạng học hỏi từ dữ liệu và ảnh hưởng đến chất lượng dự đoán của mô hình.

### 3.1.4. Iteration

Iteration (lặp) là quá trình lặp đi lặp lại một tập hợp các thao tác cho đến khi đạt được một điều kiện nhất định. Trong lập trình, iteration thường được sử dụng trong vòng lặp để thực hiện một công việc nhiều lần.

Trong Deep Learning, iteration thường đề cập đến:

* Số lần cập nhật trọng số của mô hình trong quá trình huấn luyện.
* Một lần duyệt qua batch dữ liệu trong quá trình học.

Ví dụ, nếu có 1000 mẫu dữ liệu, batch size = 100: Một epoch sẽ có 1000 / 100 = 10 iterations.

### 3.1.5. Loss

Loss là một giá trị đo lường sự khác biệt giữa dự đoán của mô hình và giá trị thực tế. Mục tiêu của học máy là giảm loss xuống mức thấp nhất có thể, nghĩa là mô hình dự đoán càng chính xác.

* Nếu loss cao: Mô hình chưa tốt, cần điều chỉnh.
* Nếu loss thấp: Mô hình hoạt động tốt.

Có nhiều loại loss function (hàm mấy mát) tùy thuộc vào bài toán. Trong bài toán này, hàm mất mát là hàm CrossEntropyLoss

### 3.1.6. Accuracy

Accuracy (độ chính xác) là tỉ lệ số mẫu được dự đoán đúng so với tổng số mẫu.

Trong một bài toán có n mẫu dữ liệu, độ chính xác được tính bằng:

* + n: Tổng số mẫu trong tập dữ liệu.
  + : Nếu dự đoán đúng thì giá trị là 1, nếu sai thì giá trị là 0.
  + : Tổng số lần mô hình đoán đúng

### 3.1.7. Overfitting và underfitting

Overfitting là khi mô hình học quá kỹ vào dữ liệu huấn luyện, đến mức nó ghi nhớ luôn cả nhiễu (noise) thay vì học được quy luật tổng quát. Điều này làm cho mô hình có độ chính xác rất cao trên tập huấn luyện nhưng kém trên dữ liệu mới. Nguyên nhân là do mô hình quá phức tạp so với dữ liệu (có quá nhiều tham số, số layer quá sâu,...).

Cách khắc phục: Thu thập thêm dữ liệu để mô hình không bị ràng buộc vào một tập dữ liệu nhỏ.

Ngược lại, underfitting là khi mô hình quá đơn giản, không đủ khả năng học được mối quan hệ quan trọng trong dữ liệu. Điều này làm cho mô hình có độ chính xác thấp trên cả tập huấn luyện và tập kiểm tra. Nguyên nhân là do mô hình quá đơn giản (ít tham số, số layer quá ít...), dữ liệu chưa được xử lý tốt hoặc không có đặc trưng quan trọng hoặc số epoch quá ít.

Cách khắc phục:

* Dùng mô hình phức tạp hơn (nhiều tham số hơn, số layer sâu hơn...).
* Huấn luyện lâu hơn (tăng số epoch).
* Sử dụng feature engineering để tạo ra đặc trưng tốt hơn.
* Giảm Regularization nếu đang dùng quá mạnh.



Hình 3.1: Minh họa underfit, fit và overfit

## 3.2. Huấn luyện mô hình

### 3.2.1. Thu thập dữ liệu

Chương trình yêu cầu dữ liệu khuôn mặt để có thể huấn luyện mô hình và nhận dạng. Ở đây em sẽ chỉ lấy dữ liệu khuôn mặt của một số bạn để thử nghiệm.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.2: Cơ sở dữ liệu khuôn mặt

Khi nhận dạng được khuôn mặt, chương trình sẽ tiến hành so sánh các dữ liệu có trong các thư mục với khuôn mặt được phát hiện trên stream. Nếu trùng với thư mục nào thì sẽ hiển thị nhãn của thư mục đó, không thì sẽ hiển thị “Unknown“.

### 3.2.2. Tiền xử lý dữ liệu

def preprocess\_image(image\_path, required\_size=(160, 160)):

image = cv2.imread(image\_path)

if image is None:

print(f"Warning: Không thể đọc ảnh từ {image\_path}")

return None

image\_rgb = cv2.cvtColor(image, cv2.COLOR\_BGR2RGB)

image\_resized = cv2.resize(image\_rgb, required\_size)

return image\_resized

Sau khi sử dụng OpenCV để đọc ảnh đầu vào, chương trình sẽ tiến hành đổi kiểu ảnh từ BGR thành RGB bằng cv2.cvtColor để VGGFace2 nhận dạng được.

Kế tiếp, ảnh sẽ được resize về kích thước chuẩn (required\_size) bằng cv2.resize.

Cuối cùng, ảnh được trả về sẽ là ảnh RGB đã được resize.

### 3.2.3. Xây dựng dataset

Khi đã có được hàm đọc ảnh, tiếp đến ta phải xây dựng một lớp dataset để phân tích ảnh.

Tạo class FaceDataset(Dataset) để duyệt qua thư mục dataset, mỗi thư mục con là một danh mục chứa ảnh của một người. Với mỗi ảnh, thực hiện tiền xử lý và lưu danh sách đường dẫn ảnh kèm nhãn tương ứng.

class FaceDataset(Dataset):

def \_\_init\_\_(self, data\_dir, transform=None):

self.data\_dir = data\_dir

self.transform = transform

self.images = []

self.labels = []

for idx, person in enumerate(os.listdir(data\_dir)):

person\_folder = os.path.join(data\_dir, person)

if os.path.isdir(person\_folder):

for image\_name in os.listdir(person\_folder):

image\_path = os.path.join(person\_folder, image\_name)

image = preprocess\_image(image\_path)

if image is not None:

self.images.append(image\_path)

self.labels.append(idx)

def \_\_len\_\_(self):

return len(self.images)

def \_\_getitem\_\_(self, idx):

image\_path = self.images[idx]

image = preprocess\_image(image\_path)

label = self.labels[idx]

if self.transform:

image = self.transform(image)

return image, label

Sử dụng torchvision.transforms để:Chuyển ảnh về dạng PIL và chuyển thành tensor có giá trị từ [-1,1] bằng chuẩn hóa (Normalize).

transform = transforms.Compose([

transforms.ToPILImage(),

transforms.ToTensor(),

transforms.Normalize(mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5])

])

### 3.2.4. Tiến hành huấn luyện

Đầu tiên, tải mô hình InceptionResnetV1 và sử dụng phiên bản pretrained="vggface2" để tận dụng trọng số đã học trước từ tập dữ liệu VGGFace2. Có thể đưa mô hình lên GPU nếu có (device='cuda').

Cấu hình huấn luyện: Sử dụng CrossEntropyLoss vì đây là bài toán phân loại. Dùng Adam optimizer với lr=0.001.

Vòng lặp huấn luyện: Lặp qua 30 epochs, trong mỗi epoch:

* Có batch\_size=5
* Forward pass: Đưa ảnh vào mô hình, lấy output.
* Tính loss: So sánh output với nhãn thực tế (criterion(outputs, labels)).
* Backward pass: Tính gradient (loss.backward()) và cập nhật trọng số (optimizer.step()).

Tính Accuracy:

* Lấy nhãn dự đoán bằng torch.max(outputs, 1).
* So sánh với nhãn thực tế để tính phần trăm đúng.

Sau khi huấn luyện xong, lưu mô hình dưới dạng ONNX để dễ dàng triển khai.

A screen shot of a computer

AI-generated content may be incorrect.

Hình 3.3: Loss và Accuracy trong quá trình huấn luyện

## 3.3. Đánh giá mô hình

Sử dụng hồi quy đa thức trên embeddings

Biến đổi đặc trưng thành đa thức (PolynomialFeatures):

* Chọn 1 chiều trong embeddings để làm feature.
* Chuyển feature thành dạng đa thức bậc 2 (degree=2).

Huấn luyện mô hình Ridge Regression (alpha=0.1) giúp giảm overfitting.

Vẽ scatter plot của labels thực tế và vẽ đường hồi quy dự đoán.

# Tạo đối tượng FaceDataset cho tập kiểm tra

test\_dataset = FaceDataset(r'dataset', transform=transform)

test\_dataloader = DataLoader(test\_dataset, batch\_size=32, shuffle=False)

# Lấy embeddings từ tập huấn luyện và kiểm tra

train\_embeddings, train\_labels = get\_embeddings(model, dataloader, device)

test\_embeddings, test\_labels = get\_embeddings(model, test\_dataloader, device)

# So sánh embeddings giữa tập kiểm tra và huấn luyện

distances = pairwise\_distances\_argmin\_min(test\_embeddings, train\_embeddings)

test\_preds = distances[0]

# Chọn 1 chiều từ embeddings làm feature

X\_train = train\_embeddings[:, 0].reshape(-1, 1)

y\_train = train\_labels

# Thay vì hồi quy đa thức tuyến tính, dùng Ridge Regression để regularization

degree = 2 # Giảm bậc đa thức để tránh overfitting

poly = PolynomialFeatures(degree=degree)

X\_poly = poly.fit\_transform(X\_train)

model\_poly = Ridge(alpha=0.1) # Ridge Regression với alpha=0.1 để giảm overfitting

model\_poly.fit(X\_poly, y\_train)

# Dự đoán nhãn

y\_pred\_poly = model\_poly.predict(X\_poly)

# Sắp xếp dữ liệu theo X\_train để vẽ đồ thị

sorted\_indices = np.argsort(X\_train[:, 0])

X\_sorted = X\_train[sorted\_indices]

y\_pred\_poly\_sorted = y\_pred\_poly[sorted\_indices]

# Vẽ biểu đồ

plt.scatter(X\_train, y\_train, color='blue', label='Actual labels')

plt.plot(X\_sorted, y\_pred\_poly\_sorted, color='green', label=f'Polynomial regression (degree={degree})')

plt.title('Polynomial Regression on Embeddings')

plt.xlabel('Embedding Feature')

plt.ylabel('Labels')

plt.legend()

plt.show()

A screen shot of a graph

AI-generated content may be incorrect.

Hình 3.4: Biểu đồ hồi quy đa thức bậc 2

* Trục X (Embedding Feature): Một chiều của vector embedding được trích xuất từ FaceNet.
* Trục Y (Labels): Nhãn của các khuôn mặt trong tập dữ liệu huấn luyện.
* Chấm xanh (Actual labels): Các điểm dữ liệu thực tế, tức là nhãn của ảnh khuôn mặt tương ứng với giá trị embedding.
* Đường màu xanh lá (Polynomial regression - degree=2): Đường hồi quy bậc 2 cố gắng mô hình hóa mối quan hệ giữa feature và nhãn.

CHƯƠNG IV: XÂY DỰNG CHƯƠNG TRÌNH

# 4.1. Thiết kế hệ thống

### 4.1.1. Phân tích chương trình

Cách thức hoạt động của chương trình có thể được miêu tả bởi biểu đồ hoạt động sau:

A diagram of a diagram

AI-generated content may be incorrect.

Hình 4.1: Biểu đồ hoạt động của chương trình

Chương trình sẽ nhận hình ảnh, sau đó sử dụng mô hình đã được huấn luyện để phân tích các đặc điểm của khuôn mặt rồi so sánh nó với các khuôn mặt trong dataset. Nếu hình ảnh giống với khuôn mặt nào thì sẽ trả về tên thư mục chứa khuôn mặt đó và gắn contour lên để dễ dàng nhận biết.

Tiếp đó, khi ấn điểm danh, chương trình sẽ:

* Nếu sinh viên chưa điểm danh: Lưu dữ liệu bao gồm MSSV, họ tên, thời gian điểm danh và trạng thái (đúng giờ, đi muộn) vào CSDL điểm danh.
* Nếu sinh viên đã điểm danh: Báo rằng sinh viên đã điểm danh.
* Nếu không nhận diện được khuôn mặt, chương trình sẽ báo lỗi.

### 4.1.2. Phân tích hệ thống

Hệ thống sẽ có 2 tác nhân chính là giảng viên và sinh viên. Chức năng và mối quan hệ giữa các tác nhân được miêu tả bởi biểu đồ ca sử dụng sau:

A diagram of a diagram

AI-generated content may be incorrect.

Hình 4.2: Biểu đồ ca sử dụng của hệ thống

Khi chạy chương trình, cửa sổ điểm danh sẽ hiện ra. Thông tin tài khoản được lưu trong CSDL của hệ thống.

* Tài khoản giảng viên sẽ là người quyết định thời gian vào lớp của tiết học. Cuối buổi học, giảng viên sẽ thông kê lại danh sách điểm danh của tiết học đó.
* Tài khoản sinh viên sẽ điểm danh trên hệ thống và hệ thống sẽ gửi thông tin về CSDL.

### 4.1.3. Cơ sở dữ liệu

Để lưu thông tin điểm danh của sinh viên, chúng ta cần sử dụng CSDL. Hệ thống này sẽ sử dụng SQLite.

CSDL gồm 4 bảng:

* Bảng in4\_sv: Lưu thông tin sinh viên

|  |  |
| --- | --- |
| **Tên cột** | **Kiểu dữ liệu** |
| MSSV | TEXT |
| Họ và tên | TEXT |
| Lớp | TEXT |
| Khoa | TEXT |

* Bảng Diem\_danh: Lưu thông tin điểm danh

|  |  |
| --- | --- |
| **Tên cột** | **Kiểu dữ liệu** |
| Id | TEXT |
| MSSV | TEXT |
| Thời gian điểm danh | DATETIME |
| Trạng thái điểm danh | TEXT |

* Bảng Thoigian\_tiet: Lưu thời gian điểm danh của tiết

|  |  |
| --- | --- |
| **Tên cột** | **Kiểu dữ liệu** |
| Id | INTEGER |
| Giờ | INTEGER |
| Phút | INTEGER |

* Bảng Login: Lưu thông tin tài khoản

|  |  |
| --- | --- |
| **Tên cột** | **Kiểu dữ liệu** |
| Id | TEXT |
| Password | TEXT |
| Loại tài khoản | TEXT |

### 4.1.4. Giao diện chương trình

Chương trình sẽ được thiết kế giao diện bằng html. Có những giao diện chính như sau:

* login: Cửa sổ đăng nhập
* index\_gv: Cửa sổ làm việc của tài khoản giảng viên
* index\_sv: Cửa sổ làm việc của tài khoản sinh viên
* diemdanh\_list: Hiển thị danh sách điểm danh

## 4.2. Kiểm thử chương trình

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.3: Cửa sổ đăng nhập

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.4: Cửa sổ index\_gv

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.5: Cửa sổ index\_sv

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.6: Điểm danh thành công

Link GitHub của chương trình:

A qr code with a dinosaur

AI-generated content may be incorrect.

CHƯƠNG V: TỔNG KẾT

## 5.1. Kết quả đã đạt được

Hệ thống nhận diện khuôn mặt được thử nghiệm với tập dữ liệu gồm 18 sinh viên, mỗi sinh viên có 3-5 ảnh trong thư mục dataset. Kết quả cho thấy: 18/18 sinh viên được nhận diện chính xác trong điều kiện ánh sáng phòng học thông thường, khoảng cách từ camera đến khuôn mặt từ 0.5 đến 1m.

Tuy nhiên, khi góc quay khuôn mặt nghiêng quá 45 độ hoặc ánh sáng yếu, mô hình sẽ gặp khó khăn trong việc khuôn mặt trong khung hình vì hình sẽ bị nhiễm muối, dẫn đến nhãn "Unknown" được gán, hoặc đôi lúc sẽ ra một người hoàn toàn khác.

## 5.2. Hướng phát triển của đề tài

Trong tương lai, em dự kiến sẽ có một số hướng phát triển như sau:

* Nâng cấp mô hình để nâng cao hiệu suất và cải thiện chất lượng nhận dạng.
* Làm cho khoảng cách Ơ-clit trở nên linh hoạt theo môi trường chứ không cố định một ngưỡng, nhằm

KẾT LUẬN

Tin học hoá trong công tác quản lý là đề tài có tính chất thực tế. Nếu đề tài này được hoàn thiện và áp dụng vào thực tế sẽ nâng cao được hiệu quả công tác quản lý, giúp cho cán bộ quản lý giảm bớt khó khăn, tránh được sai sót trong công việc đồng thời nâng cao hiệu quả công việc, giảm bớt được rất nhiều thời gian trong việc lập báo cáo thống kê, tra cứu... Tạo điều kiện phục vụ tốt hơn cho công tác quản lý điểm danh.

Do trình độ của em còn hạn chế nên đề tài của em vẫn chưa thể đáp ứng được nhu cầu thực tế, còn nhiều những thiếu sót hạn chế nhất định. Nhưng qua đó em đã rút ra rất nhiều kinh nghiệm cho bản thân về cách làm đề tài quản lý.

Em xin chân thành cảm ơn thầy Nguyễn Tuấn Linh đã giúp em hoàn thành đề tài này.

TÀI LIỆU THAM KHẢO

<https://chatgpt.com>

<https://github.com/bariarviv/VGGFace2>

<https://github.com/timesler/facenet-pytorch>

<https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76>