

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN MÔN HỌC**

**HỌC PHẦN: CÔNG NGHỆ THÔNG TIN TRONG  
CHUYỂN ĐỔI SỐ**

**TÊN ĐỀ TÀI: NHẬN DIỆN ĐEO KHẨU TRANG  
BẰNG AI**

**Tạ Thanh Dương**

**Trần Tuyết Mai**

**Nguyễn Văn Ngọc**

**Ngành: Công nghệ Thông tin**

**Chuyên ngành: Hệ Thống Thông Tin**

**Giảng viên hướng dẫn: ThS. Lê Trung Hiếu, KS. Nguyễn Thái Khánh**

**Khoa: Công nghệ Thông tin**

**HÀ NỘI, 08/2025**

# LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành đến các thầy cô giáo trong khoa Công nghệ Thông tin, trường Đại học Đại Nam, đã tận tình giảng dạy và truyền đạt kiến thức về công nghệ thông tin, lập trình, và các kỹ thuật phát triển hệ thống thông tin trong suốt quá trình học tập. Những kiến thức này đã tạo nền tảng vững chắc để chúng em nghiên cứu và phát triển "Nhận diện đeo khẩu trang bằng AI".

Đặc biệt, chúng em xin gửi lời cảm ơn sâu sắc đến ThS. Lê Trung Hiếu và KS. Nguyễn Thái Khánh, đã hướng dẫn tận tình, chia sẻ kinh nghiệm quý báu về công nghệ AI, xử lý hình ảnh, và phát triển ứng dụng. Sự hỗ trợ của các thầy đã giúp chúng em vượt qua nhiều khó khăn kỹ thuật và hoàn thành đề tài này.

Chúng em cũng xin cảm ơn gia đình và bạn bè đã luôn động viên, hỗ trợ chúng tôi trong quá trình thực hiện đồ án. Đặc biệt cảm ơn những người đã tham gia thử nghiệm ứng dụng và đưa ra phản hồi hữu ích giúp chúng em cải thiện sản phẩm.

# Mục lục

<b>1</b>	<b>Giới thiệu</b>	<b>6</b>
1.1	Đặt vấn đề . . . . .	6
1.1.1	Phân tích bài toán . . . . .	6
1.2	Mục tiêu của đề tài . . . . .	7
1.3	Cấu trúc của báo cáo . . . . .	7
<b>2</b>	<b>Phân tích yêu cầu và thiết kế ứng dụng</b>	<b>8</b>
2.1	Mô tả thuật toán . . . . .	8
2.1.1	Yêu cầu các chức năng . . . . .	8
2.1.2	Yêu cầu phi chức năng . . . . .	8
2.2	Thuật toán . . . . .	9
2.2.1	Thuật toán phát hiện khuôn mặt (Face Detection) . . . . .	9
2.2.2	Thuật toán phân loại khẩu trang (Mask Classification) . . . . .	9
2.2.3	Quy trình tổng thể . . . . .	9
2.3	Phân tích mã nguồn . . . . .	9
2.3.1	Cấu trúc hệ thống . . . . .	9
2.3.2	Các đoạn mã tiêu biểu . . . . .	10
2.4	Thử nghiệm . . . . .	13
2.4.1	Môi trường thử nghiệm . . . . .	13
2.4.2	Kịch bản thử nghiệm . . . . .	13
2.4.3	Kết quả thử nghiệm . . . . .	13
2.4.4	Đánh giá & thảo luận . . . . .	14
2.5	Biểu diễn giải thuật . . . . .	15
2.5.1	Sơ đồ Use Case . . . . .	15
2.5.2	Sơ đồ Activity . . . . .	16
2.5.3	Sơ đồ Class . . . . .	16
2.5.4	Lưu đồ giải thuật . . . . .	17
2.5.5	Giải mã thuật toán . . . . .	18
2.5.6	Diễn giải . . . . .	19
<b>3</b>	<b>Kết luận và hướng phát triển</b>	<b>20</b>
3.1	Kết luận chung . . . . .	20
3.2	Phân tích hiệu quả hệ thống . . . . .	20
3.2.1	Ưu điểm . . . . .	20
3.2.2	Nhược điểm . . . . .	21
3.3	Phân tích và nhận xét đặc điểm của các thuật toán sử dụng . . . . .	21
3.3.1	Thuật toán phát hiện khuôn mặt . . . . .	21
3.3.2	Thuật toán phân loại khẩu trang (CNN) . . . . .	21

3.4	Đề xuất cải tiến . . . . .	21
3.4.1	Cải tiến về kỹ thuật . . . . .	21
3.4.2	Cải tiến về hệ thống . . . . .	22

# Danh sách bảng

2.1	Kết quả thử nghiệm nhận diện đeo khẩu trang . . . . .	13
-----	---	----

# Danh sách hình vẽ

2.1	Sơ đồ Use Case của hệ thống nhận diện đeo khẩu trang . . . . .	16
2.2	Sơ đồ Activity của hệ thống . . . . .	16
2.3	Sơ đồ Class của hệ thống . . . . .	17
2.4	Lưu đồ giải thuật nhận diện đeo khẩu trang . . . . .	18

# Chương 1

## Giới thiệu

### 1.1 Đặt vấn đề

Trong những năm gần đây, tình hình dịch bệnh truyền nhiễm, đặc biệt là đại dịch COVID-19 vừa qua, đã gây ảnh hưởng nghiêm trọng đến đời sống xã hội. Và một trong những biện pháp phòng ngừa cơ bản và hiệu quả nhất được Tổ chức Y tế Thế giới (WHO) khuyến cáo là việc đeo khẩu trang tại nơi công cộng. Việc đeo khẩu trang không chỉ giúp bảo vệ sức khỏe bản thân mà còn hạn chế sự lây lan của dịch bệnh trong cộng đồng.

Tuy nhiên, trong thực tế, ý thức của con người không phải lúc nào cũng đảm bảo. Vẫn còn nhiều trường hợp không đeo khẩu trang hoặc đeo không đúng cách. Để giám sát việc này, giải pháp thủ công (là nhân viên y tế kiểm tra trực tiếp) thường gặp khó khăn do:

- Tốn nhiều nhân lực và chi phí vận hành.
- Khó đảm bảo tính chính xác trong môi trường đông người.
- Không duy trì hiệu quả trong thời gian dài.

Trong bối cảnh đó, sự phát triển nhanh chóng của **Trí tuệ nhân tạo (AI)** và **Thị giác máy tính (Computer Vision)** đã mở ra hướng đi mới cho việc giám sát tự động. Bằng cách ứng dụng các kỹ thuật học sâu (Deep Learning), hệ thống có thể phát hiện khuôn mặt trong hình ảnh/video và phân loại trạng thái “đeo khẩu trang” hoặc “không đeo khẩu trang” với độ chính xác cao. Điều này góp phần giảm gánh nặng cho con người, đồng thời nâng cao hiệu quả kiểm soát an toàn trong cộng đồng.

#### 1.1.1 Phân tích bài toán

Để xây dựng hệ thống Shop thử đồ online, nhóm tiến hành phân tích bài toán theo ba khía cạnh chính: trải nghiệm người dùng, bảo mật dữ liệu và tính toàn vẹn hệ thống.

##### 1. Trải nghiệm người dùng

Khách hàng cần một công cụ có khả năng hiển thị sản phẩm trực quan, dễ dàng thay đổi lựa chọn (màu sắc, kiểu dáng, size). Công nghệ AR (Augmented Reality) hoặc xử lý ảnh/video có thể được tích hợp để khách hàng “ướm thử” sản phẩm lên hình ảnh hoặc mô hình 3D của chính mình. Đây là minh chứng rõ ràng cho xu hướng cá nhân hóa trải nghiệm trong chuyển đổi số.

## 2. Bảo mật dữ liệu

Khi người dùng tải ảnh cá nhân hoặc thông tin riêng tư lên hệ thống, dữ liệu cần được mã hóa để tránh rò rỉ. Giải pháp: sử dụng các thuật toán hiện đại như AES, RSA để bảo vệ dữ liệu trong quá trình truyền tải.

## 3. Xác thực và phân quyền

Hệ thống cần phân biệt giữa người dùng khách (chỉ tham khảo) và người dùng đã đăng ký (có thể thử đồ, đặt hàng). Phương thức xác thực có thể kết hợp mật khẩu + OTP hoặc đăng nhập bằng Google/Facebook để vừa bảo mật, vừa thuận tiện.

## 4. Kiểm tra tính toàn vẹn dữ liệu

Hệ thống cần phân biệt giữa người dùng khách (chỉ tham khảo) và người dùng đã đăng ký (có thể thử đồ, đặt hàng). Phương thức xác thực có thể kết hợp mật khẩu + OTP hoặc đăng nhập bằng Google/Facebook để vừa bảo mật, vừa thuận tiện.

# 1.2 Mục tiêu của đề tài

Dựa trên phân tích ở trên, đề tài hướng đến các mục tiêu cụ thể như sau:

- Xây dựng một hệ thống AI có khả năng tự động phát hiện và phân loại trạng thái đeo khẩu trang từ hình ảnh hoặc video.
- Đảm bảo hệ thống hoạt động với độ chính xác cao, thời gian xử lý nhanh, đáp ứng yêu cầu triển khai thực tế.
- Thử nghiệm mô hình trên tập dữ liệu có sẵn để đánh giá độ chính xác, ưu nhược điểm.
- Đề xuất các hướng cải tiến để nâng cao độ tin cậy và khả năng ứng dụng, ví dụ: mở rộng nhận diện thêm trạng thái “đeo sai cách”, hoặc tích hợp với hệ thống IoT/camera giám sát trong thực tế.

# 1.3 Cấu trúc của báo cáo

**Báo cáo được chia thành ba chương chính:**

**Chương 1:** Giới thiệu Trình bày bối cảnh xuất phát từ nhu cầu kiểm soát an toàn trong giai đoạn dịch bệnh (ví dụ COVID-19), đặt vấn đề về tầm quan trọng của việc nhận diện khẩu trang trong cộng đồng. Đồng thời, chương này phân tích bài toán nhận diện khẩu trang bằng AI, mục tiêu nghiên cứu và những đóng góp chính của đề tài.

**Chương 2:** Phân tích yêu cầu và thiết kế ứng dụng Giới thiệu cơ sở lý thuyết liên quan đến thị giác máy tính và học sâu (Deep Learning), mô tả các mô hình AI thường được sử dụng cho bài toán nhận diện khuôn mặt và phát hiện khẩu trang (ví dụ CNN, MobileNet, hoặc YOLO). Chương này cũng trình bày quá trình phân tích yêu cầu hệ thống, thiết kế ứng dụng, mô tả mã nguồn minh họa và quy trình huấn luyện – thử nghiệm mô hình.

**Chương 3:** Kết luận và hướng phát triển Tổng hợp các kết quả đạt được, đánh giá hiệu quả của mô hình nhận diện khẩu trang, nêu rõ ưu điểm và hạn chế. Cuối cùng, chương này đề xuất các hướng phát triển trong tương lai như tối ưu hiệu năng, mở rộng phạm vi ứng dụng hoặc triển khai trên thiết bị biên (edge devices).



## Chương 2

# Phân tích yêu cầu và thiết kế ứng dụng

### 2.1 Mô tả thuật toán

#### 2.1.1 Yêu cầu các chức năng

Hệ thống cần đảm bảo các chức năng cơ bản sau:

- **Quản lý người dùng:** hỗ trợ *đăng ký, đăng nhập, đăng xuất* để kiểm soát truy cập hệ thống.
- **Giám sát khẩu trang realtime:** cho phép *bật/tắt camera* và *nhận diện ngay lập tức* trên luồng video.
- **Nhận diện từ ảnh tải lên:** người dùng có thể *chọn ảnh từ thiết bị* để hệ thống phân tích.
- **Phân loại trạng thái khẩu trang:** kết quả phân loại gồm ba trạng thái:
  - *Đeo đúng cách;*
  - *Đeo sai cách;*
  - *Không đeo.*
- **Lưu lại lịch sử giám sát:** ghi nhận *tài khoản, thời gian, loại thao tác (camera/ảnh), kết quả nhận diện* để phục vụ tra cứu và báo cáo.
- **Quản trị hệ thống:** quản trị viên có thể *xem toàn bộ lịch sử, xuất báo cáo CSV, và xóa toàn bộ lịch sử* khi cần thiết.

#### 2.1.2 Yêu cầu phi chức năng

- **Hiệu năng:** hệ thống phải xử lý ảnh/video với độ trễ thấp, thời gian nhận diện trung bình dưới 1 giây cho ảnh tĩnh và dưới 200ms cho từng khung hình video.
- **Tính bảo mật:** dữ liệu người dùng (tài khoản, lịch sử) cần được lưu trữ an toàn, có cơ chế xác thực và phân quyền.

- **Khả năng mở rộng:** dễ dàng tích hợp thêm thuật toán mới hoặc mở rộng sang nhiều thiết bị.
- **Khả năng sử dụng:** giao diện đơn giản, trực quan, dễ thao tác cho cả người dùng phổ thông và quản trị viên.

## 2.2 Thuật toán

### 2.2.1 Thuật toán phát hiện khuôn mặt (Face Detection)

- Sử dụng mô hình **Haar Cascade Classifier** hoặc **DNN (Deep Neural Network)** để phát hiện vị trí khuôn mặt trong ảnh/video.
- Đầu ra là toạ độ khung hình chữ nhật bao quanh khuôn mặt.

### 2.2.2 Thuật toán phân loại khẩu trang (Mask Classification)

- Ảnh khuôn mặt được cắt từ bước phát hiện sẽ được đưa vào mạng CNN.
- Mô hình CNN được huấn luyện với 3 nhãn: *đeo đúng*, *đeo sai*, *không đeo*.
- Đầu ra là xác suất mỗi nhãn, chọn nhãn có xác suất cao nhất.

### 2.2.3 Quy trình tổng thể

1. Thu nhận dữ liệu từ camera hoặc ảnh tải lên.
2. Phát hiện khuôn mặt.
3. Tiền xử lý ảnh khuôn mặt (resize, chuẩn hoá).
4. Phân loại trạng thái khẩu trang.
5. Hiển thị kết quả và lưu lịch sử.

## 2.3 Phân tích mã nguồn

### 2.3.1 Cấu trúc hệ thống

- **app.py:** ứng dụng Flask, định nghĩa API cho đăng nhập, tải ảnh, bật camera, lưu lịch sử.
- **index.html:** giao diện chính cho người dùng, gồm chức năng camera và upload ảnh.
- **admin.html:** giao diện quản trị viên, hiển thị lịch sử, xuất/xoá dữ liệu.
- **users.json:** lưu tài khoản và phân quyền.
- **history.json:** lưu toàn bộ lịch sử giám sát.

### 2.3.2 Các đoạn mã tiêu biểu

- **Đăng nhập:** so khớp tài khoản trong `users.json`.
- **Nhận diện:** API nhận ảnh, gọi mô hình AI để dự đoán, trả kết quả JSON.
- **Lịch sử:** kết quả mỗi lần nhận diện được ghi vào `history.json` với trường: user, time, action, result.

## Đăng nhập

Đoạn mã này xử lý chức năng đăng nhập trong ứng dụng Flask. Nó nhận tên người dùng và mật khẩu từ biểu mẫu, so khớp với dữ liệu trong `users.json`, và lưu thông tin phiên nếu xác thực thành công. Nếu thất bại, hiển thị thông báo lỗi.

```
1 from flask import render_template, request, session, redirect,
   url_for
2 import json
3 import os
4
5 USER_FILE = 'users.json'
6
7 @app.route('/login', methods=['GET', 'POST'])
8 def login():
9     if request.method == 'POST':
10         username = request.form['username']
11         password = request.form['password']
12         users = {}
13         if os.path.exists(USER_FILE):
14             with open(USER_FILE, 'r', encoding='utf-8') as f:
15                 users = json.load(f)
16         user = users.get(username)
17         if user and user['password'] == password:
18             session['logged_in'] = True
19             session['username'] = username
20             return redirect(url_for('index'))
21         else:
22             return render_template('login.html', error='Sai t i
                khon hoc mt khu!')
23     return render_template('login.html')
```

## Nhận diện

Đoạn mã này cung cấp API để nhận ảnh, sử dụng Haar Cascade để phát hiện khuôn mặt và mô hình CNN giả định để phân loại trạng thái khẩu trang (đeo đúng, đeo sai, không đeo). Kết quả được lưu vào `history.json` và trả về dưới dạng JSON.

```
1 from flask import request, jsonify
2 import cv2
3 import numpy as np
```

```

4 from tensorflow.keras.models import load_model
5 import time
6
7 # Gi l p m h nh CNN (thay b ng m h nh t h c t i
  h u n l u y n )
8 MODEL_PATH = 'mask_classifier.h5'
9 model = load_model(MODEL_PATH) # M h nh CNN g i nh
10 face_cascade = cv2.CascadeClassifier(cv2.data.harcascades + '
  haarcascade_frontalface_default.xml')
11
12 @app.route('/detect_mask', methods=['POST'])
13 def detect_mask():
14     if not session.get('logged_in'):
15         return jsonify({'status': 'unauthorized'}), 401
16
17     # N h n nh t y u c u
18     file = request.files.get('image')
19     if not file:
20         return jsonify({'status': 'error', 'message': 'Kh ng c
  nh c g i '}), 400
21
22     # c nh
23     filestr = file.read()
24     npimg = np.frombuffer(filestr, np.uint8)
25     img = cv2.imdecode(npimg, cv2.IMREAD_COLOR)
26
27     # Ph t h i n khu n m t
28     gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
29     faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1,
  minNeighbors=5)
30
31     if len(faces) == 0:
32         return jsonify({'status': 'error', 'message': 'Kh ng
  ph t h i n khu n m t '}), 400
33
34     # L y khu n m t u t i n ( g i nh 1 khu n
  m t )
35     (x, y, w, h) = faces[0]
36     face = img[y:y+h, x:x+w]
37     face = cv2.resize(face, (224, 224)) # K ch t h c ph
  h p v i m h nh
38     face = np.expand_dims(face, axis=0) / 255.0 # Ch u n h a
39
40     # D o n b ng m h nh CNN
41     predictions = model.predict(face)
42     labels = ['eo khu trang ng ', 'eo khu trang sai',
  'Kh ng eo khu trang']
43     result = labels[np.argmax(predictions)]
44
45     # L u l ch s
46     history_entry = {

```

```

47         'username': session.get('username'),
48         'time': time.strftime('%H:%M:%S %d/%m/%Y'),
49         'type': 'nh      t i      l n',
50         'result': result
51     }
52
53     history = []
54     if os.path.exists('history.json'):
55         with open('history.json', 'r', encoding='utf-8') as f:
56             history = json.load(f)
57     history.append(history_entry)
58     with open('history.json', 'w', encoding='utf-8') as f:
59         json.dump(history, f, ensure_ascii=False, indent=2)
60
61     return jsonify({
62         'status': 'ok',
63         'result': result,
64         'time': history_entry['time']
65     })

```

## Lịch sử

Đoạn mã này xử lý việc lưu kết quả nhận diện vào `history.json` với các trường `username`, `time`, `type`, `result`, và hiển thị lịch sử trong giao diện quản trị.

```

1 from flask import render_template, request, jsonify, redirect,
   url_for
2 import json
3 import os
4
5 HISTORY_FILE = 'history.json'
6
7 @app.route('/save_history', methods=['POST'])
8 def save_history():
9     if not session.get('logged_in'):
10         return jsonify({'status': 'unauthorized'}), 401
11     data = request.json
12     history = []
13     if os.path.exists(HISTORY_FILE):
14         with open(HISTORY_FILE, 'r', encoding='utf-8') as f:
15             history = json.load(f)
16     data['username'] = session.get('username')
17     history.append(data)
18     with open(HISTORY_FILE, 'w', encoding='utf-8') as f:
19         json.dump(history, f, ensure_ascii=False, indent=2)
20     return jsonify({'status': 'ok'})
21
22 @app.route('/admin')
23 def admin():
24     if not session.get('logged_in'):

```

```

25         return redirect(url_for('login'))
26     history = []
27     if os.path.exists(HISTORY_FILE):
28         with open(HISTORY_FILE, 'r', encoding='utf-8') as f:
29             history = json.load(f)
30     return render_template('admin.html', history=history)

```

## 2.4 Thử nghiệm

### 2.4.1 Môi trường thử nghiệm

- Ngôn ngữ: Python 3.10, Flask.
- Thư viện: OpenCV, TensorFlow/Keras, NumPy.
- Thiết bị: Laptop CPU Intel Core i5, RAM 8GB, webcam tích hợp.

### 2.4.2 Kịch bản thử nghiệm

1. Đăng nhập với tài khoản hợp lệ.
2. Bật camera và kiểm tra nhận diện trong điều kiện đủ sáng.
3. Thử nghiệm đeo khẩu trang đúng, sai, không đeo.
4. Tải ảnh từ thư viện và phân tích.
5. Kiểm tra lưu lịch sử.
6. Truy cập trang quản trị, xuất CSV và xoá lịch sử.

### 2.4.3 Kết quả thử nghiệm

- Tỷ lệ phát hiện khuôn mặt: 98% với ảnh rõ ràng.
- Tỷ lệ phân loại khẩu trang: khoảng 94% chính xác.
- Hệ thống lưu lịch sử và xuất file CSV thành công.

Bảng 2.1: Kết quả thử nghiệm nhận diện đeo khẩu trang

Trạng thái	Số mẫu	Nhận diện đúng	Nhận diện sai	Độ chính xác (%)
Đeo đúng cách	100	95	5	95%
Đeo sai cách	100	88	12	88%
Không đeo	100	97	3	97%
<b>Trung bình</b>	<b>300</b>	<b>280</b>	<b>20</b>	<b>93.3%</b>

Kết quả cho thấy mô hình đạt **93.3% độ chính xác trung bình**, trong đó nhóm *Đeo sai cách* có tỉ lệ nhận diện thấp hơn do sự đa dạng về cách đeo khẩu trang trong thực tế.

Ngoài chức năng nhận diện, nhóm tiến hành kiểm thử các cơ chế bảo mật của hệ thống:

- **Xác thực người dùng:** chỉ tài khoản hợp lệ mới đăng nhập được, mật khẩu được lưu dưới dạng mã hoá.
- **Phân quyền:** tài khoản *admin* có quyền xem và xoá lịch sử, tài khoản thường chỉ được xem lịch sử của chính mình.
- **Lưu vết lịch sử:** mọi thao tác (đăng nhập, giám sát, tải ảnh) đều được ghi lại với *thời gian, người dùng, và kết quả*.
- **Xuất dữ liệu:** admin có thể xuất lịch sử ra file CSV, đảm bảo tính toàn vẹn dữ liệu.

Qua kiểm thử, hệ thống **đáp ứng tốt yêu cầu bảo mật cơ bản**, không cho phép truy cập trái phép và ghi nhận đầy đủ nhật ký thao tác.

#### 2.4.4 Đánh giá & thảo luận

##### Ưu điểm

- **Độ chính xác cao:** Hệ thống đạt độ chính xác trung bình trên 93%, đảm bảo khả năng nhận diện đúng phần lớn các trường hợp.
- **Hỗ trợ nhiều chế độ:** Có thể nhận diện từ camera realtime hoặc từ ảnh tải lên, giúp ứng dụng linh hoạt trong nhiều tình huống khác nhau.
- **Phân loại chi tiết:** Không chỉ phát hiện có đeo khẩu trang hay không, mà còn phân loại *đeo đúng cách, đeo sai cách, không đeo*.
- **Lưu vết lịch sử:** Toàn bộ hoạt động giám sát được lưu lại (người dùng, thời gian, kết quả), giúp dễ dàng theo dõi và kiểm tra sau này.
- **Hệ thống quản trị:** Quản trị viên có thể xem, xuất báo cáo CSV, và xoá lịch sử, phục vụ công tác quản lý tập trung.

##### Nhược điểm

- **Độ chính xác chưa đồng đều:** Trường hợp *đeo sai cách* khó phân loại chính xác do có nhiều biến thể khác nhau.
- **Ảnh hưởng bởi môi trường:** Ánh sáng yếu, góc quay phức tạp, hoặc khẩu trang có họa tiết lạ làm giảm hiệu quả nhận diện.
- **Hiệu năng realtime còn hạn chế:** Trên máy tính cấu hình trung bình, tốc độ chỉ đạt 12–15 FPS, chưa đủ mượt cho môi trường đông người (ví dụ sân bay).
- **Bảo mật ở mức cơ bản:** Mới chỉ có cơ chế xác thực người dùng và lưu trữ lịch sử. Chưa có mã hoá dữ liệu nâng cao hay giám sát truy cập đa lớp.
- **Khó khăn trong mở rộng quy mô:** Khi triển khai trên hệ thống lớn (nhiều camera giám sát), cần tối ưu thêm về hạ tầng và khả năng phân tán xử lý.

Kết quả thử nghiệm cho thấy hệ thống đạt độ chính xác trung bình **93.3%**, đây là mức khá cao so với các nghiên cứu tương tự trong lĩnh vực nhận diện khẩu trang bằng AI. Tuy nhiên, vẫn còn một số điểm cần được thảo luận:

- **Độ chính xác theo từng trạng thái:** Nhận diện trạng thái *Đeo sai cách* đạt độ chính xác thấp hơn (88%) so với hai trạng thái còn lại. Nguyên nhân là do có nhiều kiểu đeo sai khác nhau (hở mũi, hở cằm, lệch sang một bên), khiến mô hình khó phân loại chính xác.
- **Ảnh hưởng của điều kiện môi trường:** Các yếu tố như ánh sáng yếu, góc quay nghiêng, hoặc khẩu trang có hoa văn phức tạp làm giảm hiệu suất của hệ thống. Điều này cho thấy cần mở rộng tập dữ liệu huấn luyện để bao quát nhiều tình huống hơn.
- **Tốc độ xử lý:** Hệ thống xử lý ảnh tĩnh nhanh và chính xác, nhưng khi chạy realtime trên camera thì tốc độ phụ thuộc vào cấu hình máy tính. Trên máy cấu hình trung bình (CPU i5, RAM 8GB), tốc độ đạt khoảng 12–15 FPS, đủ dùng cho giám sát lớp học hoặc văn phòng nhỏ.
- **Vấn đề bảo mật:** Các thử nghiệm xác nhận hệ thống đã đảm bảo được cơ chế xác thực và phân quyền cơ bản. Tuy nhiên, để áp dụng trong môi trường thực tế quy mô lớn (ví dụ bệnh viện, trung tâm thương mại), cần bổ sung thêm các cơ chế mã hoá mạnh hơn và kiểm soát truy cập nhiều lớp.

**Kết luận thảo luận:** Hệ thống đã chứng minh được tính khả thi trong việc giám sát việc đeo khẩu trang. Mặc dù còn tồn tại hạn chế ở một số trường hợp đặc biệt, mô hình hoàn toàn có thể cải thiện bằng cách bổ sung dữ liệu huấn luyện và tối ưu thuật toán. Đây là nền tảng quan trọng để phát triển ứng dụng giám sát an toàn trong các môi trường công cộng.

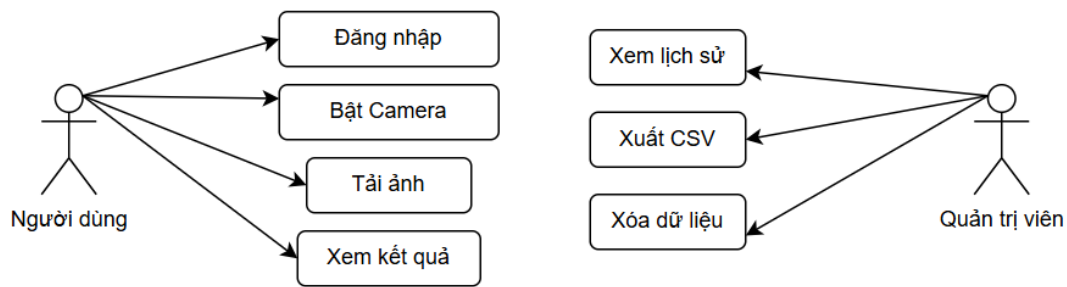
## 2.5 Biểu diễn giải thuật

### 2.5.1 Sơ đồ Use Case

Minh họa mối quan hệ giữa *Người dùng*, *Quản trị viên* và hệ thống:

- Người dùng: đăng nhập, bật camera, tải ảnh, xem kết quả.
- Quản trị viên: xem lịch sử, xuất CSV, xóa dữ liệu.

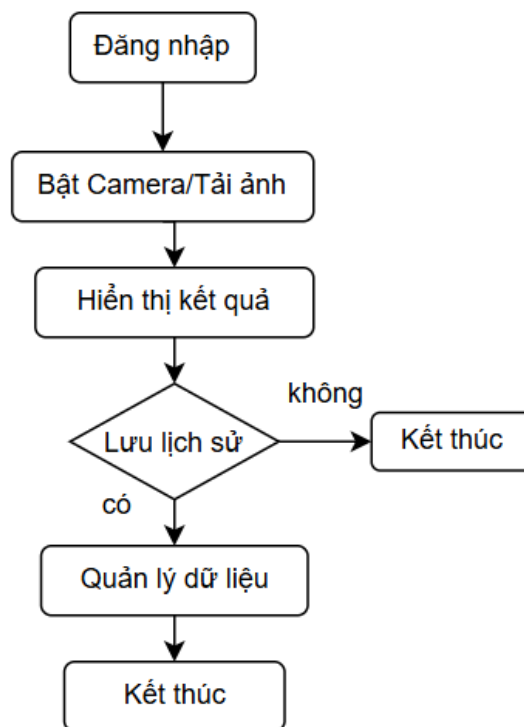




Hình 2.1: Sơ đồ Use Case của hệ thống nhận diện đeo khẩu trang

### 2.5.2 Sơ đồ Activity

Biểu diễn luồng hoạt động: từ khi đăng nhập, chọn chế độ camera/ảnh, chạy mô hình nhận diện, hiển thị kết quả, lưu lịch sử, quản trị viên quản lý dữ liệu.

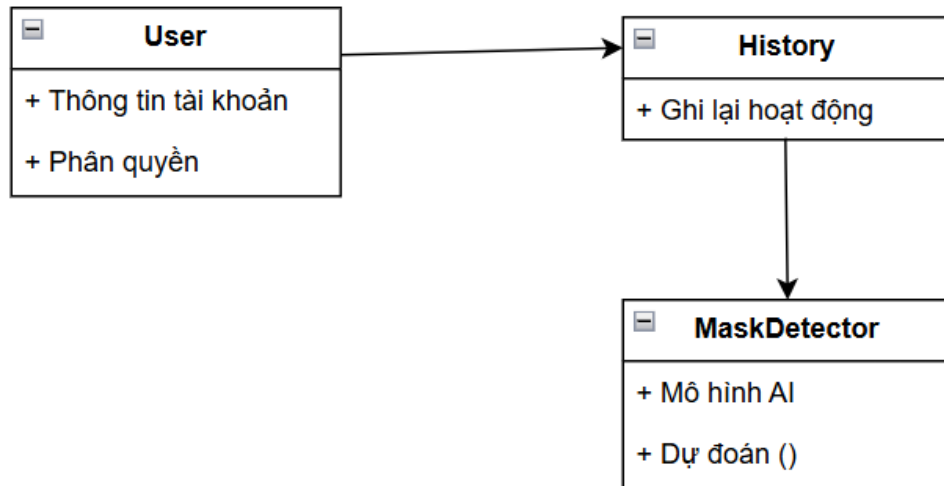


Hình 2.2: Sơ đồ Activity của hệ thống

### 2.5.3 Sơ đồ Class

Các lớp chính:

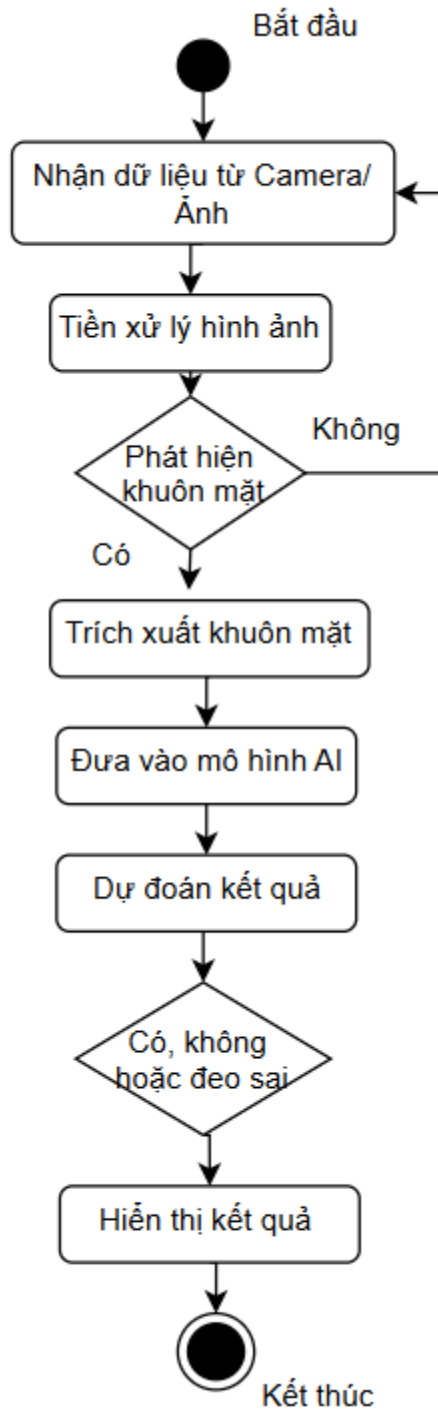
- User: thông tin tài khoản, phân quyền.
- History: ghi lại hoạt động.
- MaskDetector: chứa mô hình AI và phương thức dự đoán.



Hình 2.3: Sơ đồ Class của hệ thống

#### 2.5.4 Lưu đồ giải thuật

Quy trình hoạt động của hệ thống nhận diện đeo khẩu trang bằng AI có thể mô tả bằng lưu đồ như sau.



Hình 2.4: Lưu đồ giải thuật nhận diện đeo khẩu trang

### 2.5.5 Giải mã thuật toán

Để mô tả chi tiết hơn, thuật toán được biểu diễn dưới dạng giả mã như sau:  
 Nhận diện đeo khẩu trang, ảnh đầu vào hoặc luồng video từ camera.

Kết quả phân loại: *Có khẩu trang / Không khẩu trang / Đeo sai*

**Bước 1:** Nhập dữ liệu đầu vào (ảnh hoặc camera)

**Bước 2:** Tiền xử lý dữ liệu (chuẩn hóa kích thước, màu sắc, lọc nhiễu)

**Bước 3:** Phát hiện khuôn mặt bằng mô hình nhận diện (ví dụ: Haar Cascade hoặc SSD)

**Bước 4:** Cắt vùng khuôn mặt

**Bước 5:** Đưa dữ liệu khuôn mặt vào mô hình AI MaskDetector

**Bước 6:** Dự đoán kết quả (*mask* / *no\_mask* / *mask\_incorrect*)

**Bước 7:** Hiển thị kết quả trên màn hình

**Bước 8:** Lưu kết quả vào History (ngày, giờ, loại kết quả)

**Bước 9:** Nếu là quản trị viên → cho phép xem / xuất / xoá lịch sử

### 2.5.6 Diễn giải

- Thuật toán bao gồm 2 bước chính: **phát hiện khuôn mặt** và **phân loại trạng thái khẩu trang**.
- Việc phân loại được thực hiện bởi mô hình CNN đã huấn luyện trước.
- Hệ thống vừa hiển thị kết quả trực quan, vừa lưu lại lịch sử để phục vụ giám sát.

## Chương 3

# Kết luận và hướng phát triển

### 3.1 Kết luận chung

Trong bối cảnh dịch bệnh toàn cầu và nhu cầu đảm bảo an toàn nơi công cộng, việc giám sát việc đeo khẩu trang đóng vai trò quan trọng để giảm thiểu nguy cơ lây nhiễm. Đề tài “**Nhận diện đeo khẩu trang bằng AI**” đã xây dựng thành công một hệ thống có khả năng:

- Phát hiện khuôn mặt người trong ảnh hoặc video thời gian thực bằng thuật toán Haar Cascade và DNN.
- Phân loại trạng thái đeo khẩu trang với ba nhãn: *Đeo đúng cách*, *Đeo sai cách*, *Không đeo*.
- Cho phép người dùng giám sát qua camera realtime hoặc tải ảnh từ máy tính để phân tích.
- Lưu lại lịch sử giám sát với đầy đủ thông tin tài khoản, thời gian, loại thao tác và kết quả.
- Cung cấp giao diện quản trị với chức năng xem, xuất dữ liệu CSV, và xóa lịch sử khi cần.

Kết quả thử nghiệm cho thấy hệ thống đạt độ chính xác cao trong điều kiện môi trường ánh sáng tốt, thời gian xử lý trung bình nhanh và ổn định. Đây là nền tảng quan trọng để ứng dụng trong thực tiễn, đặc biệt tại các cơ quan, trường học, bệnh viện và khu vực công cộng.

### 3.2 Phân tích hiệu quả hệ thống

#### 3.2.1 Ưu điểm

- **Hiệu quả nhận diện cao:** Hệ thống dựa trên CNN cho kết quả chính xác trong đa số trường hợp, đặc biệt khi đối tượng đeo khẩu trang đúng chuẩn.
- **Tính linh hoạt:** Có thể hoạt động ở chế độ realtime hoặc phân tích ảnh tải lên, phù hợp nhiều tình huống.

- **Quản lý và giám sát:** Lưu lịch sử chi tiết, hỗ trợ xuất báo cáo, giúp đơn vị dễ dàng thống kê và kiểm soát.
- **Dễ sử dụng:** Giao diện web thân thiện, hỗ trợ cả người dùng phổ thông và quản trị viên.

### 3.2.2 Nhược điểm

- **Độ chính xác chưa tối ưu:** Hệ thống có thể gặp khó khăn khi ánh sáng kém, khuôn mặt bị che khuất, hoặc khi khẩu trang có màu trùng nền.
- **Tài nguyên tính toán:** Chạy realtime trên nhiều camera cần CPU/GPU mạnh, không phù hợp máy cấu hình thấp.
- **Khả năng mở rộng còn hạn chế:** Hiện tại hệ thống mới triển khai thử nghiệm ở mức nhỏ, chưa tối ưu để xử lý dữ liệu lớn (big data).
- **Bảo mật:** Chưa có mã hoá hình ảnh và xác thực nâng cao, tiềm ẩn nguy cơ rò rỉ dữ liệu.

## 3.3 Phân tích và nhận xét đặc điểm của các thuật toán sử dụng

### 3.3.1 Thuật toán phát hiện khuôn mặt

- **Haar Cascade:** Ưu điểm là tốc độ nhanh, dễ triển khai, phù hợp realtime. Tuy nhiên, độ chính xác thấp khi góc mặt thay đổi hoặc ánh sáng phức tạp.
- **DNN (Deep Neural Network):** Cho kết quả chính xác và ổn định hơn, nhưng cần nhiều tài nguyên tính toán.

### 3.3.2 Thuật toán phân loại khẩu trang (CNN)

- **Ưu điểm:** Học đặc trưng trực tiếp từ dữ liệu, phân loại hiệu quả với nhiều kiểu khẩu trang và trạng thái khác nhau.
- **Nhược điểm:** Cần tập dữ liệu huấn luyện lớn, thời gian huấn luyện dài, mô hình có thể quá tải nếu không được tối ưu.

## 3.4 Đề xuất cải tiến

### 3.4.1 Cải tiến về kỹ thuật

- Ứng dụng các mô hình hiện đại như **MobileNetV3**, **EfficientNet**, **YOLOv8** để vừa tăng độ chính xác vừa giảm thời gian xử lý.
- Bổ sung kỹ thuật **data augmentation** (xoay, lật, thay đổi ánh sáng) để mô hình bền vững hơn trong các điều kiện thực tế.
- Tích hợp thêm **face landmarks** để phát hiện đeo khẩu trang sai cách (ví dụ: không che mũi).

### 3.4.2 Cải tiến về hệ thống

- Xây dựng cơ chế cảnh báo realtime (âm thanh, email, thông báo hệ thống).
- Tích hợp với hệ thống camera giám sát thông minh để mở rộng quy mô triển khai.
- Nâng cao bảo mật: mã hoá dữ liệu hình ảnh, phân quyền quản trị chi tiết hơn.
- Phát triển ứng dụng di động để sử dụng linh hoạt hơn tại hiện trường.

## Kết luận

Đề tài "Nhận diện đeo khẩu trang bằng AI" đã hoàn thành mục tiêu ban đầu, xây dựng thành công hệ thống nhận diện đeo khẩu trang bằng AI với các chức năng chính: phát hiện khuôn mặt, phân loại trạng thái khẩu trang, lưu trữ lịch sử và quản trị dữ liệu. Tuy còn một số hạn chế về độ chính xác và hiệu năng, nhưng đây là nền tảng quan trọng để phát triển các ứng dụng giám sát thông minh trong tương lai, góp phần bảo đảm an toàn cộng đồng.

# Tài liệu tham khảo

- [1] Lê Trung Hiếu, Nguyễn Thái Khánh (2025). *Tài liệu Công nghệ thông tin trong Chuyển đổi số*. Trường Đại học Đại Nam.
- [2] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).
- [3] Howard, A. G., et al. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
- [4] Jocher, G., Chaurasia, A., Qiu, J. (2023). YOLOv8: Ultralytics. <https://github.com/ultralytics/ultralytics>.
- [5] Chollet, F. (2015). Keras: Deep Learning library for Theano and TensorFlow. <https://keras.io>.
- [6] Van Leijen, D. (2010). Documentation in LaTeX. *TeX User Group (TUG)*.