**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CHẤT LƯỢNG CAO**

**LỚP 02CLC**



**BÀI BÁO VỀ GIẢI THUẬT - XÂY DỰNG**

**TRÍ TUỆ NHÂN TẠO**

**ĐỀ TÀI NHẬN DIỆN GÓC QUAY KHUÔN MẶT NGƯỜI REALTIME**

**GVHD: TS Nguyễn Trường Thịnh**

**Sinh viên thực hiện:**

**Họ và tên: Lê Quốc An.**

**MSSV: 19146142**

**TP. Hồ Chí Minh, ngày 23 tháng 6 năm 2022**

**MỤC LỤC**

[CHƯƠNG I. GIỚI THIỆU 1](#_Toc106794114)

[1.1. Giới thiệu đề tài. 1](#_Toc106794115)

[1.2. Giới thiệu phương pháp. 1](#_Toc106794116)

[CHƯƠNG II. PHƯƠNG PHÁP 3](#_Toc106794117)

[2.1. Giới thiệu về phương pháp Convolutional Neural Network. 3](#_Toc106794118)

[2.2. Tiến hành code và giải thích. 5](#_Toc106794119)

[CHƯƠNG III. KẾT QUẢ 10](#_Toc106794120)

[3.1. Lập trình Real-time. 10](#_Toc106794121)

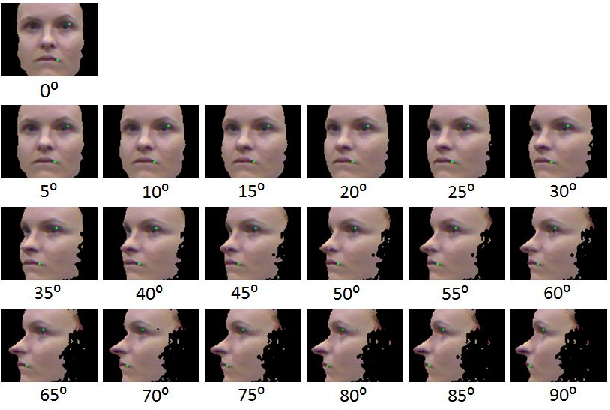
[3.2. Kết quả. 11](#_Toc106794122)

[CHƯƠNG IV. KẾT LUẬN 12](#_Toc106794123)

# **CHƯƠNG I. GIỚI THIỆU**

## **1.1. Giới thiệu đề tài.**

Phương pháp nhận dạng và hiệu chỉnh góc xoay của khuôn mặt được đề xuất có thể được sử dụng trong các hệ thống nhận dạng người bằng hình ảnh khuôn mặt bitmap, đặc biệt để bù cho góc chụp không phù hợp của ảnh tạo thành cơ sở dữ liệu khuôn mặt.



**Hình 1.1.** Nhận diện góc quay khuôn mặt.

## **1.2. Giới thiệu phương pháp.**

**Diagram

Description automatically generated with low confidence**

**Hình 1.2.** Các góc quay cần nhận diện của khuôn mặt.

Trong Project này, em sẽ xác định 3 yếu tố chính của góc quay khuôn mặt như góc quay hướng lên, hướng xuống (Pitch), góc quay trái (Yaw), phải và góc nghiêng trái, phải (Roll). Hiển thị được góc độ chính xác với độ sai số thấp.

Project này sẽ được sử dụng phương pháp CNN ( Convolutional Neural Network).

# **CHƯƠNG II. PHƯƠNG PHÁP**

**2.1. Giới thiệu về phương pháp Convolutional Neural Network.**

Mạng CNN là một tập hợp các lớp Convolution chồng lên nhau và sử dụng các hàm nonlinear activation như ReLu và tanh để kích hoạt các trọng số trong các node. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo.

Trong mô hình mạng truyền ngược (feedforward neural network) thì mỗi neural đầu vào (input node) cho mỗi neural đầu ra trong các lớp tiếp theo.

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

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

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

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

Chart

Description automatically generated

Trong mô hình CNN có 2 khía cạnh cần quan tâm là **tính bất biến** (Location Invariance) và **tính kết hợp** (Compositionality). Với cùng một đối tượng, nếu đối tượng này được chiếu theo các gốc độ khác nhau (translation, rotation, scaling) thì độ chính xác của thuật toán sẽ bị ảnh hưởng đáng kể.

Pooling layer sẽ cho bạn tính bất biến đối với phép dịch chuyển (translation), phép quay (rotation) và phép co giãn (scaling). Tính kết hợp cục bộ cho ta các cấp độ biểu diễn thông tin từ mức độ thấp đến mức độ cao và trừu tượng hơn thông qua convolution từ các filter.

Đó là lý do tại sao CNNs cho ra mô hình với độ chính xác rất cao. Cũng giống như cách con người nhận biết các vật thể trong tự nhiên.

**2.2. Tiến hành code và giải thích.**

**Bước 1:** Khai báo các Thư viện.

**Text

Description automatically generated**

**Hình 2.1.** Khai báo các thư viện cần thiết.

**Bước 2: Load Dataset và chuẩn bị khâu huấn luyện.**

**Text

Description automatically generated**

**Hình 2.2. Công đoạn Load Dataset và hiển thị đặc điểm của Dataset.**

**Table

Description automatically generated with low confidence**

**Hình 2.3.** Cài đặt các thông số cho các giá trị góc quay.

**Bước 3: Tiến hành Scale các biến đầu vào.**

Text

Description automatically generated

**Hình 2.4.** Sử dụng bộ StardardScale để scale các biến đầu vào.

**Bước 4: Tiến hành Train Model.**

Graphical user interface, text

Description automatically generated

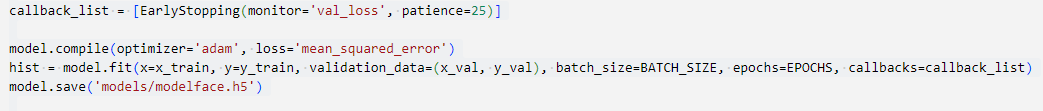
**Hình 2.5.** Set thông số Batchsize và Epoch.

Graphical user interface, text

Description automatically generated

**Hình 2.6.** Thiết lập các thông số trước khâu huấn luyện.

**Chúng ta tiến hành train và thu được kết quả :**



Text

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

**Hình 2.7.** Kết quả thu được sau khi kết thúc huấn luyện.

**Bước 4. Khảo sát Model.**

Text

Description automatically generated

Graphical user interface, chart, histogram

Description automatically generated

**Hình 2.8.** Biểu đồ mối quan hệ giữa 2 giá trị Loss Train và Loss Val.

Text

Description automatically generated

A picture containing text, antenna

Description automatically generated

A picture containing text, antenna

Description automatically generated

Chart

Description automatically generated

**Hình 2.9.** Biểu đồ biểu diễn cho 3 giá trị góc đo.

# **CHƯƠNG III. KẾT QUẢ**

## **3.1. Lập trình Real-time.**

Text

Description automatically generated

**Hình 3.1.** Thiết lập các giá trị tương thích với các hướng nhìn.

Sau đó chúng ta sử dụng lệnh PutText của Cv2 để gán các giá trị Print lên màn hình đang chạy real time. Ta thu được kết quả như bên dưới.

Text

Description automatically generated

**Hình 3.2.** Code hiển thị chữ tương thích lên màn hình OpenCv.

## **3.2. Kết quả.**

A picture containing text, person, person, indoor

Description automatically generated

A picture containing text, person, indoor

Description automatically generatedA picture containing text, person, indoor

Description automatically generated

A picture containing text, person, person, indoor

Description automatically generatedA picture containing text, person, indoor

Description automatically generated

A picture containing text, person

Description automatically generatedA picture containing text, person, indoor

Description automatically generated

**Hình 3.3.** Kết quả hướng nhìn kèm góc độ sau khi chạy thời gian thực.

# **CHƯƠNG IV. KẾT LUẬN**

Thông qua bài Project cuối kì do thầy **Nguyễn Trường Thịnh** hướng dẫn, em đã nắm được quy trình tạo và xử dữ liệu đầu vào, cũng như huấn luyện ra một sản phẩm AI. Đề tài hệ thống nhận diện góc quay khuôn mặt của em tuy độ tin cậy chưa đạt đến mức cao nhất, nhưng trong tương lai, em sẽ phát triển thêm nhiều tín năng, cũng như tăng độ chính xác của hệ thống. Chúng ta không thể phủ nhận được các lợi ích mà Trí tuệ Nhân tạo đã mang lại cho đời sống con người. Em xin gửi lời cảm ơn sâu sắc đến thầy Nguyễn Trường Thịnh và các bạn trợ giảng, đã sẵn sàng trả lời các thắc mắc, câu hỏi mà em đưa ra.

