Shape, square

Description automatically generated

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**--------------o0o--------------**

A blue logo with a black background

Description automatically generated with low confidence

**BÁO CÁO ĐỒ ÁN**

**MÔN : XỬ LÝ ẢNH VÀ ỨNG DỤNG – CS406**

ĐỀ TÀI:

FACIAL EXPRESSION RECOGNITION

NHẬN DIỆN CẢM XÚC KHUÔN MẶT

**GVHD: Cáp Phạm Đình Thăng**

**Nhóm sinh viên thực hiện: Nhóm GS**

**Thành viên**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phạm Xuân Hoàng** | |  | **20520519** | |
|  |  | | |
|  | |  |  | |

*Thành phố Hồ Chí Minh, ngày 31 tháng 1 năm 2024*

**Lời nói đầu**

Báo cáo nghiên cứu này tập trung vào việc phát triển và triển khai một hệ thống nhận diện biểu cảm khuôn mặt hiệu quả. Chúng tôi đã sử dụng mô hình học sâu để tự động phân loại biểu cảm từ các hình ảnh khuôn mặt. Qua quá trình đào tạo trên bộ dữ liệu FER2013, hệ thống của chúng tôi đã đạt được độ chính xác nhất định trong việc nhận diện nhiều biểu cảm khác nhau.

Chúng tôi đã thực hiện thử nghiệm và đánh giá hiệu suất của hệ thống trên bộ dữ liệu thực tế, đồng thời so sánh với các phương pháp khác. Kết quả cho thấy rằng hệ thống của chúng tôi có khả năng nhận diện biểu cảm khuôn mặt với độ chính xác và độ tin cậy cao.

Bên cạnh đó, báo cáo này cũng đề cập đến các thách thức gặp phải trong quá trình phát triển và triển khai hệ thống, cũng như đề xuất hướng nghiên cứu và cải tiến trong tương lai để nâng cao hiệu suất của hệ thống nhận diện biểu cảm khuôn mặt.

**Lời cảm ơn**

Sau một kì học tập bổ ích và thú vị, tập thể nhóm GS bày tỏ lòng biết ơn đến thầy Cáp Phạm Đình Thăng. Sự nhiệt huyết, tri thức được thầy chia sẻ từ cả trong lớp học và từ bên ngoài, cùng những góp ý chi tiết mà thầy đã đưa ra trong suốt quá trình thực hiện đồ án đã mang lại cho chúng tôi những kiến thức mới, trải nghiệm mới, cải thiện tư duy chuyên môn trong môn Xử lý ảnh nói riêng và lĩnh vực Thị giác máy tính nói chung. Chúng em chân thành bày tỏ lòng biết ơn sâu sắc đến thầy và chúc thầy luôn giữ vững sức khỏe, luôn tràn đầy năng lượng tích cực và duy trì niềm đam mê để tiếp tục truyền đạt kiến thức cho những thế hệ sau như thầy đã hướng dẫn chúng tôi trong kỳ học này!

**Vai trò**

|  |  |  |
| --- | --- | --- |
| **Tên thành viên** | **MSSV** | **Vai trò** |
| **Phạm Xuân Hoàng** | **20520519** | **Tìm dataset**  **Lý thuyết**  **Cài đặt source code**  **Viết báo cáo** |

***Chân thành cảm ơn sự đóng góp của các thành viên trong đồ án này***

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*……., ngày……...tháng……năm 20…*

**Người nhận xét**

*(Ký tên và ghi rõ họ tên****)***

**Mục lục**

[**1.** **Giới thiệu đề tài** 4](#_Toc157637463)

[**1.1** **Xác định bài toán** 5](#_Toc157637464)

[**1.2** **Phạm vi nghiên cứu** 5](#_Toc157637465)

[**1.3** **Ứng dụng** 6](#_Toc157637466)

[**2.** **Các hướng tiếp cận** 6](#_Toc157637467)

[**2.1** **Phương pháp truyền thống** 6](#_Toc157637468)

[**2.2** **Phương pháp hiện đại** 7](#_Toc157637469)

[**3.** **Methodology** 7](#_Toc157637470)

[**3.1 Face Detector** 8](#_Toc157637471)

[***a.*** **MTCNN (Multi-task Cascaded Convolutional Networks)** 8](#_Toc157637472)

[***b.*** **Mediapipe** 10](#_Toc157637473)

[**3.2** **Preprocessing** 11](#_Toc157637474)

[***a.*** **Quá trình training** 11](#_Toc157637475)

[***b.*** **Quá trình inference** 12](#_Toc157637476)

[**3.3 CNN Model** 12](#_Toc157637477)

[***a.*** **ResNet50** 12](#_Toc157637478)

[***b.*** **Classifier** 12](#_Toc157637479)

[**4.** **Experiment** 13](#_Toc157637480)

[**4.1** **Dataset** 13](#_Toc157637481)

[**4.2** **Cài đặt** 14](#_Toc157637482)

[**4.3** **Result** 14](#_Toc157637483)

[**4.4** **Nhận xét và hướng phát triển** 16](#_Toc157637484)

[**5.** **Conclusion** 17](#_Toc157637485)

[Reference 18](#_Toc157637486)

# **Giới thiệu đề tài**

Đối với đa số mọi người, biểu cảm trên khuôn mặt là một trong những tín hiệu mạnh mẽ, tự nhiên và phổ biến nhất để con người truyền tải trang thái cảm xúc và ý nghĩ. Ekman và Friesen [1] đã chỉ ra rằng con người nhận thức được một số cảm xúc cơ bản theo cùng một cách bất kể nền tảng văn hóa hay quốc gia nào và họ đã xác định có sáu loại cảm xúc cơ bản bao gồm: Giận dữ, ghê tởm, sợ hãi, vui vẻ, buồn bã và ngạc nhiên. Trong một nghiên cứu mở rộng khác, Ekman và Heider [2] đã bổ sung thêm một loại cảm xúc nữa là khinh bỉ. Ngoài ra, FER2013 [3], một bộ cơ sở dữ liệu quy mô lớn được giới thiệu trong IMCL 2013, cũng giới thiệu và phân loại các khuôn mặt với bảy loại trạng thái cảm xúc khác nhau bao gồm: Giận dữ, ghê tởm, sợ hãi, vui vẻ, buồn bã, ngạc nhiên và bình thường (**Hình 1**). Trong các nghiên cứu khác, các nhà khoa học cũng đã giới thiệu nhiều loại mô hình khác nhau để cung cấp nhiều loại cảm xúc hơn do sự phức tạp của nét mặt. Tuy nhiên, các cảm xúc mở rộng này chiếm một phần khá nhỏ trong các biểu hiện cảm xúc hàng ngày nên chưa được đưa vào trong nghiên cứu này.

A screenshot of a computer game

Description automatically generated

***Hình 1:*** *7 loại cảm xúc cơ bản của con người : Tức giận, vui vẻ, buồn bã, ngạc nhiên, sợ hãi, kinh tởm, bình thường (Theo FER2013)*

### **Xác định bài toán**

**Input**: **Ảnh** chứa khuôn mặt người.

**Output**: **Dòng** **text** biểu thị cảm xúc của khuôn mặt từ ảnh đầu vào.

A group of people in a stadium

Description automatically generated

***Hình 2:*** *Mô tả input (trái) và output (phải) của bài toán*

### **Phạm vi nghiên cứu**

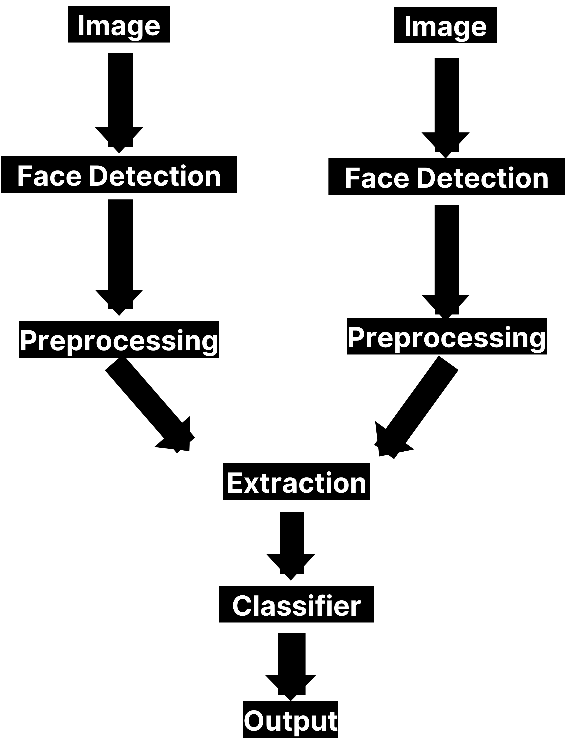
Trong đồ án này chúng tôi nghiên cứu và tìm cách sử dụng ngôn ngữ Python để nhận diện 7 trạng thái cảm xúc : **Happy** (vui vẻ), **Sad** (buồn bã), **Angry** (tức giận), **Surprise** (ngạc nhiên), **Fear** (sợ hãi), **Disgust** (kinh tởm), **Neutral** (bình thường).

### **Ứng dụng**

* Kinh doanh : Phân tích phản ứng của khách hàng khi tiếp nhận và sử dụng sản phẩm, dịch vụ.
* Y tế : Theo dõi sức khỏe tinh thần bệnh nhân.
* Giáo dục : Quản lý, chăm sóc tâm lý học đường.
* Trí tuệ nhân tạo : Tăng cường khả năng giao tiếp của các người máy.
* …

# **Các hướng tiếp cận**

### **Phương pháp truyền thống**



***Hình 3:*** *Minh họa các bước thực hiện phương pháp truyền thống, quá trình training (trái) và quá trình inference (phải)*

Ý tưởng của phương pháp này như sau: đưa ảnh chứa khuôn mặt vào tiền xử lý, sau đó rút trích đặc trưng bằng Facial Landmark + HOG feature [4], rồi đưa vào phân loại bằng các thuật toán máy học như Linear Regression, SVM,…

Nhiều nghiên cứu và thực nghiệm đi trước chỉ ra rằng SVM [5] là thuật toán mang lại hiệu quả cao nhất cho cách tiếp cận này. Quá trình tiền xử lý của quá trình huấn luyện (training) và suy luận (inference) sẽ khác nhau nên chúng tôi tách **Hình 3** thành 2 nhánh riêng cho 2 quá trình.

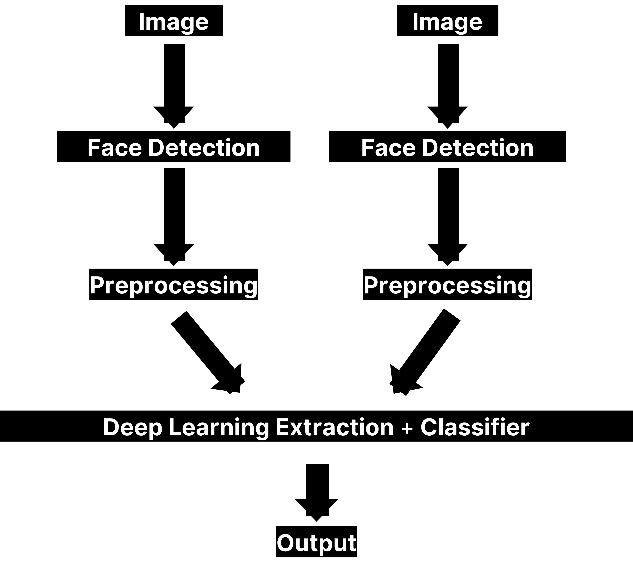
**Ưu điểm :**

* Dễ cài đặt.
* Hiệu suất khá tốt.

**Nhược điểm :**

* + - Ảnh hưởng ánh sáng, nhiễu và bị che khuất khiến hiệu suất mô hình kém đi.

### **Phương pháp hiện đại**



***Hình 4:*** *Minh họa các bước thực hiện phương pháp hiện đại, quá trình training (trái) và quá trình inference (phải)*

Ở phương pháp hiện đại, các bước thực hiện tương tự như phương pháp truyền thống, tuy nhiên người ta cho rằng các phương pháp máy học cũng như cách lấy đặc trưng khuôn mặt cũ không hiệu quả bằng các mô hình học sâu, và thay vì sử dụng các phương pháp máy học và độc lập 2 quá trình rút trích đặc trưng và phân loại thì một mô hình học sâu (thường là CNN) sẽ đảm nhận cả 2 công việc này, tạo thành 1 mô hình end-to-end. Quá trình training và inference cũng được tách ra thành 2 nhánh như phương pháp truyền thống.

**Ưu điểm :**

* Hiệu suất cao hơn.
* Việc xử lý các yếu tố môi trường trở nên hiệu quả hơn rất nhiều.

**Nhược điểm :**

* + - Khó cài đặt hơn

Vậy thì phương pháp nào hiệu quả hơn? Để giải quyết, người ta đã tổ chức các cuộc thi để thúc đẩy sự tiến bộ trong lĩnh vực này. Trong số đó, có một cuộc thi dùng database FER2013 [6], các phương pháp hiện đại đã cho thấy hiệu quả vượt trội hơn trong bài toán nhận dạng cảm xúc khuôn mặt. Vì lý do đó, bài báo cáo này sẽ tập trung nghiên cứu vào phương pháp theo hướng hiện đại.

# **Methodology**

Theo như phương pháp hiện đại đã đề cập ở 2.2, chúng tôi sẽ đi nghiên cứu một số phương pháp cụ thể cho từng bước : Face Detection (Nhận dạng khuôn mặt), Preprocessing (Tiền xử lý dữ liệu), CNN model (mô hình CNN để rút trích đặc trưng và phân loại cảm xúc).

## **3.1 Face Detector**

Face Detector có nhiệm vụ phát hiện và lấy khuôn mặt trong ảnh. Trong bài báo cáo này chúng tôi tìm hiểu về 2 Face Detector là MTCNN và Mediapipe

### **MTCNN (Multi-task Cascaded Convolutional Networks)**

* Mạng MTCNN [7] được giới thiệu vào năm 2015, gồm 3 lớp FCN chồng lên nhau : Proposal Network (P-Net), Refine Network (R-Net), Output Network (O-Net). Mỗi mạng có cấu trúc khác nhau và đảm nhiệm vai trò khác nhau.

A collage of a person's face

Description automatically generated

***Hình 5:*** *Mạng MTCNN*

+ Đầu tiên áp dụng Image Pyramid, tạo một loạt các bản sao với nhiều kích cỡ khác nhau của ảnh, sau đó dùng kernel trượt qua toàn bộ trên từng bản sao để thu được các bản quan sát, sau đó đưa các bản quan sát này vào mạng P-Net.

+ P-Net có nhiệm vụ nhận bản quan sát, trả về các bounding box của những vị trí được cho là khuôn mặt (bounding box candidates, xin viết tắt là *candidates*) trên bản quan sát, sau đó áp dụng thuật toán Bounding Box Regression (kỹ thuật để dự đoán vị trí của bounding box khi chúng ta cần phát hiện đối tượng) và Non-Maximum Suppression ( gộp bounding box bị chồng chéo ). Sau cùng mạng trả về là các box (đã được normalize về khoảng (0,1)) và điểm confident score của candidates tương ứng.

A diagram of a diagram of a diagram

Description automatically generated with medium confidence

***Hình 6****: P-Net*

+ Các candidates sau khi gộp sẽ được đưa qua R-Net. R-Net có nhiệm vụ khá giống P-Net, nhưng có thêm 1 lớp padding để thêm các zero pixel vào bounding box, giúp tinh chỉnh tọa độ, xử lý các bounding box trong trường hợp bị vượt quá biên ảnh, tiếp tục thực hiện Non-Maximum Suppression và Bounding Box Regression để gộp các candidates lại.

A diagram of a network connection

Description automatically generated

***Hình 7:*** *R-Net*

+ Đầu ra của R-Net lại được đưa vào O-Net và thực hiện tương tự R-Net, nhưng có thực hiện thêm Facial Localization, khiến đầu ra của O-Net không chỉ có bounding box mà còn có thêm tọa độ landmark (2 mắt, mũi, 2 cánh môi) cùng confidence score của nó.

A diagram of a graph

Description automatically generated with medium confidence

***Hình 8:*** *O-Net*

**Ưu điểm :**

* Có thể phát hiện được khuôn mặt ở các góc độ khó (ánh sáng, góc độ mặt,…).
* Dễ cài đặt.
* Có paper riêng và nhiều nguồn để tìm hiểu.
* Hiệu suất khá tốt.

**Nhược điểm :**

* Tốc độ chậm, đặc biệt là khi sử dụng CPU.
* Khá lỗi thời vì đã ra đời từ khá lâu.

### **Mediapipe**

MTCNN đã ra đời từ 2015, và quá trình thực nghiệm của chúng tôi cho thấy rằng tốc độ của MTCNN là quá chậm, nên chúng tôi đã tìm tới các Face Detector có tốc độ cao hơn như HaarCascade [8], DLib [9],YOLOv5 [10]… Kế hoạch ban đầu chúng tôi muốn sử dụng YOLOv5, nhưng trong báo cáo lần này, chúng tôi muốn thử nghiệm cài đặt và sử dụng một Face Detector khác có hiệu suất ở mức ổn, nhưng tốc độ rất ấn tượng mặc dù không cần sử dụng GPU, đó là Mediapipe.

Mediapipe được phát triển bởi Google, không chỉ có các tác vụ về khuôn mặt mà còn cả Hand Landmark, Facial Landmark, Pose Estimation, Object Detection, các tác vụ NLP, Audio,…Đối với bài toán Face Detection, Mediapipe đưa ra giải pháp sử dụng mạng BlazeFace.

A table with numbers and symbols

Description automatically generated with medium confidence A diagram of a program

Description automatically generated

***Hình 9:*** *Một kiến trúc extractor của BlazeFace (trái) và Single BlazeBlock cùng Double BlazeBlock (phải)*

BlazeFace [11] được lấy ý tưởng từ việc cải tiến cấu trúc của các mô hình MobileNet, ngoài việc đề xuất sử dụng Single BlazeBlock và Double BlazeBlock cùng với việc kết hợp với SSD framework tối ưu cho việc inference trên mobile GPUs, Blazeface cũng thay thế Non-Max Suppression bằng một chiến lược là **Blending Strategy** như sau :

1. Nếu không có box nào trùng nhau thì bỏ qua
2. Nếu có nhóm box trùng nhau thì gộp lại thành **weightedbox** lại theo phương pháp sau:

**weightedbox** có 4 điểm tọa độ là tổng các tích tọa độ box[i] với score[i] rồi chia cho tổng score. Ở dạng công thức sẽ là như sau :

+ Confidence score của **weightedbox** là trung bình các confident score của các box trùng.

Nhờ chiến lược này mà tốc độ xử lý của Blazeface rất nhanh, thực nghiệm so sánh với MobileNetV2-SSD [12] là nhanh hơn gấp 2.3 lần, một thống kê rất ấn tượng.

Trên trang của Mediapipe có đề cập rằng mô hình có sử dụng Blazeface, nhưng lại thay thể backbone của nó bằng một backbone khác, được giới thiệu là một SSD-Encoder Custom, được Mediapipe giới thiệu rằng có tốc độ có thể lên tới 275 FPS với GPU, tuy nhiên chúng tôi không tìm được tài liệu cụ thể về kiến trúc này. Cùng với đó, một số tài liệu khác lại đề cập rằng Mediapipe đã chọn bounding box có confident score lớn nhất thay vì tính toán như trong paper của BlazeFace.

**Ưu điểm :**

* Nhanh. Hỗ trợ tốt với cả các thiết bị không có GPU.
* Hiệu suất tốt. Khả năng nhận diện trong điều kiện khá khó (ánh sáng, góc độ mặt,…).
* Có paper riêng và nhiều nguồn để tìm hiểu.
* Dễ cài đặt.

**Nhược điểm :**

* Có khá ít tài liệu để tìm hiểu.
* Hiệu suất thấp hơn vì phải đánh đổi tốc độ.
* Gặp vấn đề khi số lượng khuôn mặt nhiều hay cách quá xa.

Quá trình thực nghiệm chúng tôi thu được FPS của Mediapipe là 60 - 70 (CPU) và (70 - 120) (GPU), còn MTCNN là 10 - 20 (CPU) và 20 - 30 (GPU), cho thấy rằng Mediapipe phù hợp hơn MTCNN, thế nên chúng tôi sử dụng Mediapipe làm Face Detector cho bài toán này.

## **Preprocessing**

### **Quá trình training**

Ở quá trình Training, chúng tôi sử dụng ảnh khuôn mặt được cắt ra từ Face Detector, áp dụng các biện pháp Data Augmentation (tăng cường ảnh) như xoay, tăng giảm độ sáng, xê dịch ảnh,… làm gia tăng thêm độ đa dạng cho dữ liệu ảnh, đảm bảo mô hình có thể làm việc tốt đối với các ảnh với điều kiện môi trường khó. Sau khi áp dụng Data Augmentation, chúng tôi đưa ảnh về kích thước 197x197 để chuẩn bị đưa vào mô hình để huấn luyện.

### **Quá trình inference**

Ở quá trình Training, chúng tôi sử dụng ảnh khuôn mặt được cắt ra từ Face Detector, sau đó chúng tôi đưa ảnh về kích thước 197x197 để chuẩn bị đưa vào mô hình để suy luận.

## **3.3 CNN Model**

Chúng tôi sử dụng các model CNN đã loại bỏ lớp Fully Connected, hoặc có thể cắt tỉa thêm một vài lớp, kết hợp với một bộ Classifier gồm các Dense layers để cho đầu ra là xác suất dự đoán trên các lớp.

### **ResNet50**

ResNet50 là một mô hình mạng nơ-ron sâu (deep neural network) thuộc họ ResNet [13] (Residual Networks) được giới thiệu bởi Microsoft Research. Mô hình này có 50 lớp (layers) và nổi bật với cấu trúc "residual" (dư thừa), giúp giải quyết vấn đề đào tạo mô hình học sâu mà vẫn tránh được hiện tượng Gradient Vanishing [14] (triệt tiêu đạo hàm). ResNet50 thường được sử dụng trong các nhiệm vụ nhận diện hình ảnh và phân loại, và đã đạt được hiệu suất cao trên nhiều bộ dữ liệu thử nghiệm.

A diagram of a number of objects

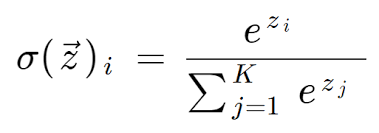
Description automatically generated with medium confidence

***Hình 10:*** *Kiến trúc ResNet50*

Chúng tôi loại bỏ lớp Softmax và dùng đầu ra của mô hình để lấy vector 2048 chiều mà mô hình trả về, làm vector đặc trưng của ảnh.

### **Classifier**

Sau khi lấy được đặc trưng của ảnh bằng ResNet50, chúng tôi đó cho đi qua thêm 3 lớp Dense sử dụng hàm kích hoạt ReLu và số unit lần lượt là 4096, 2048 và 1024, cuối cùng qua 1 lớp Softmax để tiến hành phân loại và thu được 1 list là xác suất các nhãn. Chúng tôi chọn xác suất cao nhất để trả ra output.



***Hình 11:*** *Công thức Softmax*

# **Experiment**

### **4.1 Dataset**

**FER2013** : Bộ dữ liệu là file .csv, lưu giữ hơn 36,000 ảnh grayscale 48x48 chứa mặt người, mỗi ảnh được gán số 1 trong 7 nhãn biểu thị cảm xúc tương ứng như sau: Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral. Chúng tôi sử dụng data từ 1 cuộc thi trên trang Kaggle [15]. Đây là bộ dataset khá thử thách, vì có khá nhiều ảnh bị gán nhãn sai, có ảnh thậm chí không có khuôn mặt.

**A screenshot of a graph

Description automatically generated**

***Hình 12:*** *Dataset FER2013 (Kaggle)*

Cấu trúc của bộ dữ liệu gồm các mẫu dữ liệu được biểu diễn thành từng hàng, với các yếu tố sau như sau :

* + emotion : cảm xúc của khuôn mặt trong mẫu dữ liệu, được đánh số tương ứng (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral)
  + Usage : vai trò của mẫu dữ liệu (Training : huấn luyện, PrivateTest, PublicTest : kiểm thử và lấy điểm cho mô hình trong cuộc thi). Chúng tôi lấy các Usage là Training để làm tập Train, Private Test làm tập Valuation, Public Test làm tập Test. Qua đó thu được tỷ lệ Train:Test:Val=8:1:1.
  + pixels : các ảnh grayscale được lưu giữ dưới dạng ma trận 2 chiều.

### **Cài đặt**

Mô hình Resnet50-based được chúng tôi thử nghiệm bằng model build sẵn lấy từ outsource keras-vggface [16] trên Github. Outsource cũng có hỗ trợ pretrained model đã huấn luyện trên tập VGGFace.

Chúng tôi thực nghiệm việc huấn luyện bằng notebook trên Kaggle, có sử dụng GPU NVIDIA TESLA P100 được Kaggle cung cấp, dùng Adam với tốc độ học khởi đầu là 0.001 làm optimizer, hàm loss là Categorical Entropy, áp dụng thêm một Learning Rate Scheduler từ Keras, cho phép giảm tốc độ học xuống mức 20% so với tốc độ học cũ mỗi khi validation accuracy không tăng sau khoảng 5 epochs để tránh các vấn đề về overfitting trong quá trình huấn luyện. Số epoch được chúng tôi đặt là 100, tuy nhiên nếu nếu learning rate giảm xuống dưới ngưỡng 0.0001 thì sẽ lập tức dừng quá trình huấn luyện. Độ đo được chúng tôi sử dụng là Accuracy.

### **Result**

Ngoài ResNet50 trong kiến trúc mô hình đã đề cập tới trong phần 3.3, chúng tôi thử nghiệm với nhiều backbone CNN khác như MobileNetV1 [17], MobileNetV2 [18], MobileNetV3 [19], đồng thời so sánh với một số phương pháp khác được công bố trên trang PaperWithCode [20]:

|  |  |
| --- | --- |
| Phương pháp | Accuracy |
| ResMaskingNet (SOTA) [21] | 76.82 |
| EmoNext [22] | 76.1 |
| VGG-19 [23] | 75.9 |
| VGG [24] | 73.3 |
| **Our** | **71.9** |
| MobileNet (Highest is MobileNetV1) | 66.5 |
| MobileNet + Attention [25] | 65.5 |

***Bảng 1:*** *So sánh với các phương pháp trên Paperwithcode*

Trong các phương pháp trên chúng tôi có đề cập tới MobileNet + Attention (Patt-Lite), mô hình mà chúng tôi đã giới thiệu trong quá trình báo cáo trực tiếp, tuy nhiên trong quá trình thực nghiệm, chúng tôi gặp khó khăn trong việc tái hiện lại kết quả của nhóm tác giả (92.5%), thay vào đó chỉ thu được 65.5%. Có thể có sự sai khác nào đó trong quá trình huấn luyện của chúng tôi, hoặc sự khác nhau về quá trình kiểm thử.

Quá trình thực nghiệm của chúng tôi diễn ra trong vòng 9 tiếng. Chúng tôi kiểm thử trên tập PrivateTest của cuộc thi và thu được kết quả **71.9%** cho độ đo Accuracy, với FPS vào khoảng 12-14 trong quá trình inference. Dưới đây mà các hình ảnh chúng tôi tổng kết từ quá trình huấn luyện:

A graph of a curve

Description automatically generatedA graph of a line

Description automatically generated with medium confidence

***Hình 13:*** *Quá trình huấn luyện mô hình với Accuracy(trái) và Loss(phải)*

Dưới đây mà các hình ảnh chúng tôi thu được khi thử nghiệm mô hình lên dữ liệu thực tế :

* Ngẫu nhiên ảnh trên tập FER2013 :

A close-up of a person's face

Description automatically generatedA close up of a person's face

Description automatically generatedA close up of a person's face

Description automatically generated

***Hình 14:*** *Kết quả nhận diên trên một số ảnh trong tập dữ liệu FER2013*

* Ngẫu nhiên ảnh thực tế :

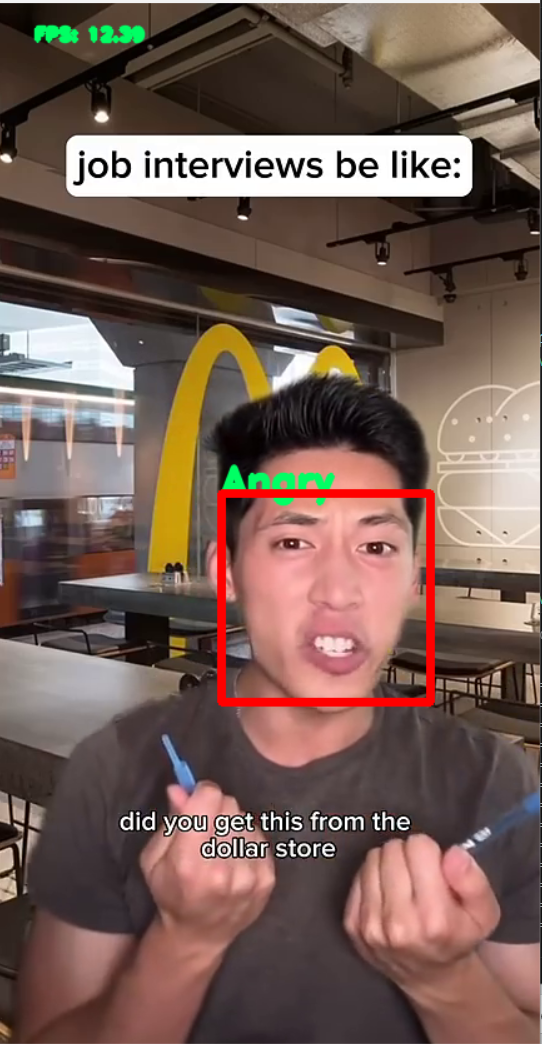
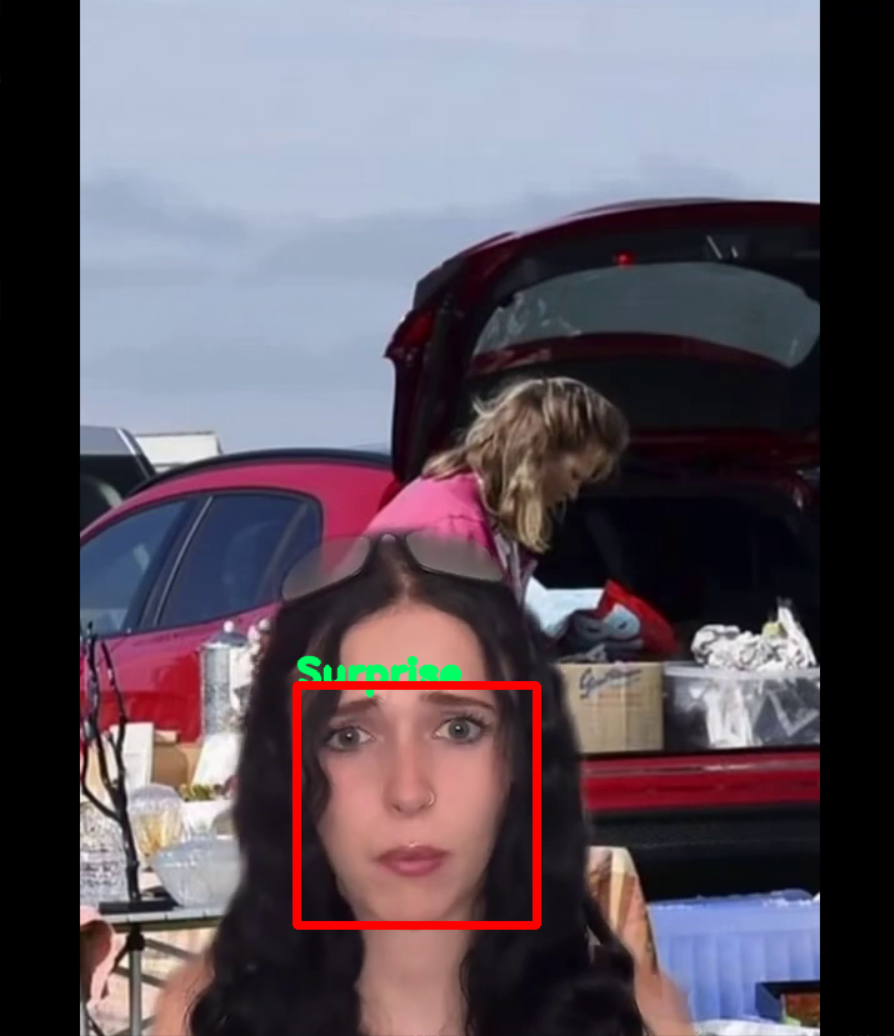
A person holding a trophy

Description automatically generated A person holding a trophy

Description automatically generated

***Hình 15:*** *Kết quả nhận diên trên một số ảnh ngẫu nhiên từ Google Images*

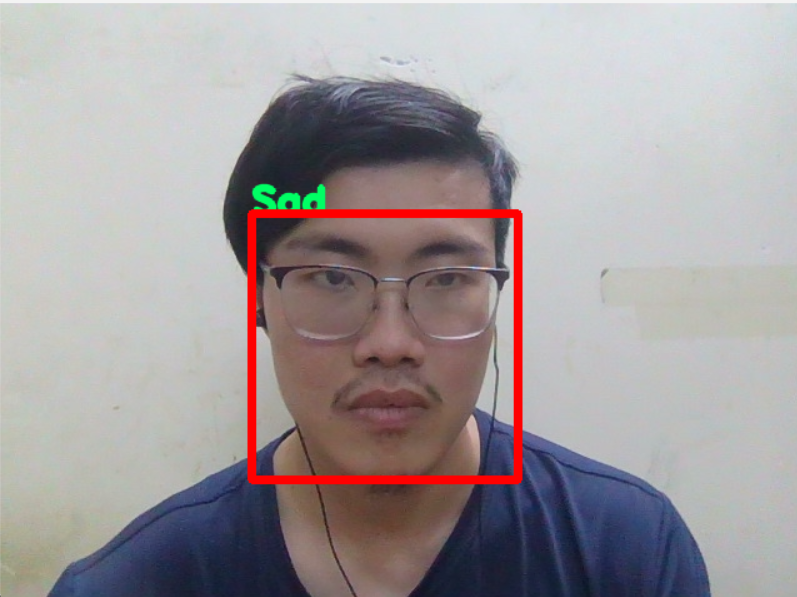
* Ngẫu nhiên video thực tế :



***Hình 16:*** *Kết quả nhận diện trên một số video trên Youtube*

* Real-time camera :

A person smiling for a picture

Description automatically generated 

***Hình 17:*** *Kết quả nhận diện trên real-time camera*

### **Nhận xét và hướng phát triển**

A blue squares with white text

Description automatically generated

***Hình 18:*** *Confusion matrix tạo ra từ việc dự đoán toàn bộ tập test*

#### **Nhận xét**

Chúng tôi đánh giá Face Detector chưa thực sự hiệu quả trong bài toán này, vì đáp ứng được việc chạy ở CPU nhưng khả năng tính toán của mô hình phân loại cảm xúc sẽ làm cho nó chậm đi đáng kể, vì thế những Face Detector chạy GPU trong bài toán này sẽ hiệu quả hơn. Vấn đề này khiến Mediapipe mất vị thế trước các mô hình khác như YOLOv5.

Chúng tôi đánh giá việc huấn luyện là chưa tốt. Kết quả nhận diện thực tế và camera cho thấy rằng mô hình hoạt động khá tốt đối với cảm xúc Happy, Angry. Tuy nhiên đang có sự nhầm lẫn giữa Neutral, Sad và Surprise, và gần như vô tác dụng với Disgust. Việc nhận diện trên webcam diễn ra không tốt lắm với 4 loại cảm xúc này, tuy nhiên khi nhận diện bằng video hay ảnh, kết quả lại tốt hơn khi nhận diện trên webcam. Có lẽ đó là lý do accuracy của chúng tôi đạt ngưỡng 71.9%. Tuy nhiên đây thật sự là một dataset khá khó, với State-of-the-art accuracy chỉ khoảng 76.82%.

Thế nên chúng tôi đã sửa đổi nhãn một chút trong quá trình inference, và đem lại kết quả tốt hơn rất nhiều.

#### **Hướng phát triển:**

* Thay thế Face Detector: Chúng tôi nhận thấy các đặc điểm của Mediapipe có thể đáp ứng trong một số môi trường nhất định (ít người, camera gần,…). Chúng tôi đang tiến hành nghiên cứu và cài đặt YOLOv5 như một giải pháp đánh đổi giữa tốc độ và hiệu suất mang tính thực tế và hiệu quả hơn.
* Thay thế cấu trúc mô hình ResNet50 của chúng tôi bằng các kiến trúc mạnh mẽ hơn, và chấp nhận đánh đổi tốc độ thực thi để lấy hiệu suất : Chúng tôi đang hướng tới một model mạnh hơn và cũng sử dụng ResNet50 như TinaFace [26], Poly-NL [27],..
* Chuyển qua một dataset khác dễ dàng hơn : FER2013 có khá nhiều những điểm nhiễu (gán nhãn sai, không có mặt,…). Chúng tôi có thể chấp nhận từ bỏ và thử sức trên một data dễ dàng hơn như **CK+** [28] (dataset này gồm hơn 900 ảnh, cũng có định dạng dữ liệu thô giống như FER2013 và có thêm nhãn “Contempt” (khinh bỉ), có State-of-the-art đạt 95%) hay **FER+** [29] (dataset này là phiên bản cải tiến của FER2013, được đội ngũ Microsoft đánh nhãn lại với định dạng 10 nhãn 1 ảnh, tương ứng với 10 người đánh nhãn cho 1 ảnh, có State-of-the-art đạt 95.55%).
* Phát triển thành các ứng dụng với mục đích cụ thể hơn như đã nói ở phần 1.3 .

# **Conclusion**

Nhận diện cảm xúc khuôn mặt là một bài toán khá thú vị và có nhiều ứng dụng thực tế. Trong bài báo cáo này, chúng tôi đã sử dụng phương pháp ResNet50-based cho bài toán Nhận diện cảm xúc khuôn mặt và thu được kết quả ở mức khá (71.9%) so với các phương pháp khác trong bộ dữ liệu. Việc thử sức với bài toán này cùng dataset ở một cuộc thi đã mang lại cho chúng tôi những trải nghiệm, kinh nghiệm và kiến thức mới. Chúng tôi cũng đề ra một số hướng để cải thiện và phát triển cho kiến trúc mô hình để mang lại hiệu quả cao hơn.

# Reference

[1] <https://www.scribd.com/document/541041918/Ekman-1971>

[2] <https://link.springer.com/article/10.1007/BF00993116>

[3] [6] [15] <https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data>

[4] <https://medium.com/nerd-for-tech/facial-landmark-detection-made-simple-7fcd079c09f0>

[5] <https://scikit-learn.org/stable/modules/svm.html>

[7] <https://arxiv.org/abs/1604.02878>

[8] <https://docs.opencv.org/3.4/d2/d99/tutorial_js_face_detection.html>

[9] <https://pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/>

[10] <https://arxiv.org/abs/2105.12931>

[11] <https://arxiv.org/abs/1907.05047>

[12]<https://www.researchgate.net/publication/346194258_MobilenetSSDv2_An_Improved_Object_Detection_Model_for_Embedded_Systems>

[13] <https://arxiv.org/abs/1512.03385>

[14] <https://medium.com/@amanatulla1606/vanishing-gradient-problem-in-deep-learning-understanding-intuition-and-solutions-da90ef4ecb54>

[16] <https://github.com/rcmalli/keras-vggface>

[17] [18] [19] <https://viblo.asia/p/cnn-architecture-series-1-mobilenets-mo-hinh-gon-nhe-cho-mobile-applications-1VgZvJV1ZAw>

[20] <https://paperswithcode.com/sota/facial-expression-recognition-on-fer2013>

[21] <https://paperswithcode.com/paper/facial-expression-recognition-using-residual>

[22] <https://paperswithcode.com/paper/emonext-an-adapted-convnext-for-facial>

[23] <https://paperswithcode.com/paper/a-novel-facial-emotion-recognition-model>

[24] <https://paperswithcode.com/paper/facial-expression-recognition-using>

[25] <https://paperswithcode.com/paper/patt-lite-lightweight-patch-and-attention>

[26] <https://paperswithcode.com/paper/tinaface-strong-but-simple-baseline-for-face>

[27] <https://paperswithcode.com/paper/poly-nl-linear-complexity-non-local-layers>

[28] <https://paperswithcode.com/sota/facial-expression-recognition-on-ck>

[29] <https://paperswithcode.com/sota/facial-expression-recognition-on-fer-1>