ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN

MÔN: XỬ LÝ ẢNH & ỨNG DỤNG

ĐỀ TÀI

Nhận Diện Cảm Xúc Khuôn Mặt

Giảng Viên : TS. Mai Tiến Dũng

Nhóm sinh viên thực hiện:

Ngô Thành Phát - 19521994

Nguyễn Quang Tuấn - 19522474

Đỗ Hoàng Phúc - 19522028

TP. Hồ Chí Minh, tháng 1 năm 2023

Tóm tắt

Trong dự án này, chúng tôi muốn xác định biểu hiện trên khuôn mặt trong một hình ảnh nhất định bằng cách sử dụng các kiến trúc học sâu . Mục tiêu là phân loại từng hình ảnh khuôn mặt thành một trong bảy loại cảm xúc trên khuôn mặt được xem xét trong nghiên cứu này. Chúng tôi đã sử dụng trích suất đặc trưng từ PCA và phương pháp học máy SVM làm đường cơ sở cho chúng tôi. Và chúng tôi đã đề xuất một phương pháp tốt hơn để xác định các biểu cảm trên khuôn mặt bằng cách sử dụng hình ảnh màu xám từ bộ dữ liệu trên trang web Kaggle. Chúng tôi đã phát triển các mô hình của mình trong Torch và sử dụng GPU ngẫu nhiên từ Google Colab. Ngoài ra chúng tôi sẽ thực hiện khái quát hóa cho bài toán bằng cách sử dụng thuật toán Viola Jones cho vấn đề nhận diện khuôn mặt. Các kết quả thí nghiệm đã cho thấy hiệu quả của các cấu trúc sâu đối với các tác vụ phát hiện biểu cảm khuôn mặt đã cải thiện một chút hiệu suất của phát hiện so với các phương pháp đường cơ sở.

Mục lục

Tóm tắt 2
LỜI CẢM ƠN 4
PHẦN 1. GIỚI THIỆU ĐỀ TÀI 5
1.1 Tổng quan 5
1.2 Ngữ cảnh ứng dụng 6
PHẦN 2. PHƯƠNG PHÁP 6
2.1 PCA 7
2.1.1 Ứng dụng PCA trong nhận dạng khuôn mặt 7
2.1.2 Học máy hỗ trợ vector SVM 9
2.1.3 Kết quả thực nghiệm 9
2.2 VGG-16 9
2.2.1 Cấu hình: 10
2.2.2 Kiến trúc: 11
2.3 Resnet-50 12
2.3.1 Kiến Trúc 14
2.4 Viola-Jones 16
2. Bộ Dữ Liệu 20
Phần 4: THỰC NGHIỆM 21
4.1 PCA và Học Máy SVM 21
4.2 VGG-16 22
4.3 Resnet-50 25
4.2 Kết Quả 26
TÀI LIỆU THAM KHẢO 28

LỜI CẢM ƠN

Đầu tiên, nhóm chúng em xin gửi lời cảm ơn chân thành đến quý thầy cô giảng viên Trường Đại học Công nghệ thông tin – Đại học Quốc gia TP. Hồ Chí Minh đã giúp cho nhóm chúng em có những kiến thức cơ bản làm nền tảng để thực hiện đề tài này. Đặc biệt, nhóm chúng em xin gửi lời cảm ơn và lòng biết ơn sâu sắc nhất tới thầy giáo – ThS Mai Tiến Dũng, thầy đã là người hướng dẫn chúng em những kiến thức cần thiết để thực hiện, cũng như đã giúp đỡ chúng em khi gặp khó khăn trong thời gian thực hiện đồ án. Nhóm em chân thành cảm ơn thầy và chúc thầy dồi dào sức khỏe.

Trong thời gian một học kỳ thực hiện đề tài, nhóm chúng em đã vận dụng những kiến thức nền tảng đã tích lũy đồng thời kết hợp với việc học hỏi và nghiên cứu những kiến thức mới từ thầy cô, bạn bè cũng như nhiều nguồn tài liệu tham tham khảo. Từ đó, nhóm chúng em vận dụng tối đa những gì đã thu nhập được để hoàn thành một đồ án tốt nhất. Tuy nhiên, vì kiến thức chuyên môn còn hạn chế và bản thân còn nhiều thiếu xót kinh nghiệm thực tiễn nên nội dung của báo cáo không tránh khỏi những thiếu xót, nhóm em rất mong nhận được sự góp ý, chỉ bảo thêm của quý thầy cô nhằm hoàn thiện những kiến thức của mình để nhóm chúng em có thể dùng làm hành trang thực hiện tiếp các đề tài khác trong tương lai cũng như là trong việc học tập và làm việc sau này.

PHẦN 1. GIỚI THIỆU ĐỀ TÀI

1.1 Tổng quan

Con người tương tác với nhau chủ yếu thông qua lời nói, nhưng cũng thông qua các cử chỉ cơ thể, để nhấn mạnh một số phần của lời nói của họ và thể hiện cảm xúc. Một trong những cách quan trọng của con người thể hiện cảm xúc là thông qua các biểu cảm trên khuôn mặt là một phần rất quan trọng của giao tiếp. Mặc dù không có gì được nói bằng lời nói, có nhiều điều cần hiểu về các thông điệp chúng tôi gửi và nhận thông qua việc sử dụng giao tiếp phi ngôn ngữ. Biểu cảm khuôn mặt truyền đạt không tín hiệu bằng lời nói, và nó đóng một vai trò quan trọng trong quan hệ giữa các cá nhân. Nhận dạng tự động các biểu cảm khuôn mặt có thể là một thành phần quan trọng của giao diện-máy tự nhiên; Nó cũng có thể được sử dụng trong khoa học hành vi và trong thực hành lâm sàng. Mặc dù con người nhận ra biểu cảm khuôn mặt hầu như không cần nỗ lực hay sự chậm trễ, nhưng nhận dạng biểu hiện đáng tin cậy bằng máy vẫn là một thách thức. Đã có một số tiến bộ trong vài năm qua về phát hiện khuôn mặt, cơ chế trích xuất tính năng và các kỹ thuật được sử dụng để phân loại biểu hiện, nhưng sự phát triển của một hệ thống tự động hoàn thành nhiệm vụ này là khó khăn [6]. Trong bài báo này, chúng tôi trình bày một cách tiếp cận dựa trên các mạng thần kinh tích chập (CNN) để nhận dạng biểu hiện trên khuôn mặt. Đầu vào vào hệ thống của chúng tôi là một hình ảnh; Sau đó, chúng tôi sử dụng CNN để dự đoán nhãn biểu hiện khuôn mặt phải là một nhãn hiệu này: tức giận, hạnh phúc, sợ hãi, buồn bã, ghê tởm và bình thường.

1.2 Ngữ cảnh ứng dụng

- Tiếp thị: Đây là một cách tuyệt vời để các công ty kinh doanh phân tích cách khách hàng phản hồi với quảng cáo, sản phẩm, bao bì và thiết kế cửa hàng của họ.
- Dịch vụ khách hàng: Quản lý dịch vụ khách hàng có thể hiệu quả hơn bằng cách sử dụng hệ thống nhận dạng cảm xúc khuôn mặt. Phân tích phản hồi của khách hàng và phản ứng của máy tính sẽ đảm bảo tương tác máy tính với con người trong cuộc sống thực.
- Hệ thống nhận diện cảm xúc khuôn mặt được sử dụng nhiều trong cuộc sống: điều trị y tế, giao tiếp song ngôn ngữ, đánh giá đau của bệnh nhân, phát hiện nói dối, giám sát trạng thái của người lái xe phát hiện trạng thái buồn ngủ dựa vào cảm xúc trên khuôn mặt được phát triển để cảnh báo cho người lái xe khi thấy dấu hiệu buồn ngủ, mệt mỏi
- Giáo dục: Nhận ra cảm xúc và sự tập trung của người học là một yếu tố quan trọng để khiến cả lớp đạt được kết quả tốt trong các hoạt động giảng dạy và học tập. Với mô hình học tập trực tuyến, việc nắm bắt cảm xúc của người học thông qua khuôn mặt được coi là một phương pháp hiệu quả để xác định mức độ quan tâm trong bài học hoặc sự tập trung của người học

PHẦN 2. PHƯƠNG PHÁP

Trong phần này, đầu tiên chúng ta sẽ sử dụng trích xuất đặc trưng PCA và học máy SVM làm đường cơ sở vấn đề phát hiện biểu cảm khuôn mặt. Và đề xuất phương pháp có khả năng tốt hơn để giải quyết vấn đề này, chúng tôi đã sử dụng một mô hình nơ-ron tích chập VGG-16 và mô hình Resnet- 50 sẽ được giới thiệu và phân tích chi tiết. Đặc biệt, chúng tôi sẽ thể hiện những lợi thế ấn tượng của nó so với phương pháp cơ sở của chúng ta

2.1 PCA

Giảm chiều dữ liệu trong machine learning là quá trình giảm thiểu số lượng đặc trưng biểu diễn dữ liệu. Việc này có thể được thực hiện theo hướng lựa chọn các đặc trưng quan trọng hoặc trích xuất các đặc trưng mới từ các đặc trưng đã có. Giảm chiều dữ liệu hữu ích trong các trường hợp như trực quan hóa, lưu trữ và năng lực tính toán hạn chế. Trong bài này hãy cùng tìm hiểu hai phương pháp giảm chiều dữ liệu nổi tiếng là PCA

PCA là viết tắt của Principal Component Analysis, có nghĩa là phân tích thành phần chính. Ý tưởng của PCA là tạo ra các đặc trưng mới độc lập là kết hợp tuyến tính của các đặc trưng cũ. Các đặc trưng mới định nghĩa một hình chiếu của dữ liệu lên một không gian con sao cho khoảng cách giữa hình chiếu và dữ liệu gốc là nhỏ nhất. Nói một cách khác, PCA tìm kiếm một không gian tuyến tính tốt nhất để xấp xỉ dữ liệu thông qua hình chiếu của nó

2.1.1 Ứng dụng PCA trong nhận dạng khuôn mặt

Mục đích:

Mục tiêu của phương pháp PCA là "giảm số chiều" của một tập vector sao cho vẫn đảm bảo được " tối đa thông tin quan trọng nhất" phương pháp PCA sẽ giữ lại K thuộc tính "mới" từ M các thuộc tính ban đầu (K<M)

Giả sử ta có N ảnh khuôn mặt, là tập ảnh huấn luyện $X1, X2, \ldots XN$

Biểu diễn mỗi ảnh thành ma trận Mx1 có dạng:

$X = (i_{i1}, i_{i2}, \ldots\ i_{iM}\ với\ i = 1, \ldots N)$

Bước 1: tính vector khuôn mặt trung bình của tập ảnh huấn luyện

$$\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$$

Bước 2: tính vector độ lệnh của mỗi khuôn mặt so với vector khuôn mặt

trung bình

$\theta_i = X_i$ - $\bar{X}$ với i = 1,…N

Bước 3: Tạo thành ma trận MxN

$$A = [\theta_1 \theta_2 \theta_N]$$

Sau đó tính ma trận hiệp phương sai M x N

$$C = \frac{1}{N} A.\bar{A}$$

Bước 4: tính các giá trị riêng của ma trận hiệp phương sai C ta được $\lambda 1$, $\lambda 2$, … λK, K≪ M Với K được tính theo công thức:

$$\frac{\sum_{i=1}^{K} \lambda 1}{\sum_{i=1}^{N} \lambda 1} \geq nguong\ (e.g\ 0.90\ or\ 0.95)$$

Bước 5: tính đặc vector riêng của ma trận hiệp phương sai C

$\psi 1, \psi 2, .. \psi K$ với $\psi_i = \frac{\psi_i}{\|\psi_i\|}\ i = 1, \dots K$

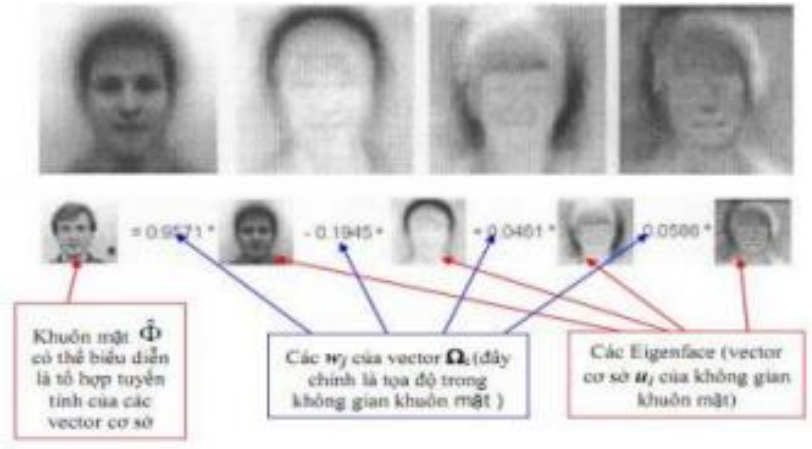
Tính K vector riêng của ma trận C theo công thức:

ψi = Av1

Bước 6: Giảm số chiều, chỉ giữ lại những thuộc tính tương ứng với các giá trị riêng lớn nhất (biểu diễn ảnh khuôn mặt trong không gian mới với K đặc trưng quan trọng nhất)

Trong không gian mới, với các vector cơ sở là mỗi ảnh khuôn mặt trong tập huấn luyện được biểu diễn thành tổ hợp tuyến tính của các vevtor cơ sở

Biểu diễn các ảnh theo vector trị riêng vừa tìm đươc. Các ảnh sẽ tương ứng với một vector trọng số wj mà mỗi hệ số của vector là hệ số tương ứng với một vector đặc trưng trong số các vector đặc trưng vừa tìm được. ta có thể biểu diễn như sau:



Đầu vào của PCA là các vector cột có M thành phần biểu diễn ảnh trong tập huấn luyện, đầu ra là các vector cột có K thành phần biểu diễn ảnh đã được trích rút đặc trưng.

2.1.2 Học máy hỗ trợ vector SVM

Phân lớp: Bước nhận dạng hay phân lớp tức là xác định danh tính (identity) hay nhãn của ảnh (label) – đó là ảnh của ai. Ở bước nhận dạng/phân lớp, ta sử dụng phương pháp SVM (Support Vector Machine). SVM sẽ tiến hành phân lớp ảnh trong tập huấn luyện, khi đưa ảnh vào nhận dạng sẽ được so sánh, tìm ra ảnh đó thuộc vào lớp nào.

2.1.3 Kết quả thực nghiệm

2.2 VGG-16

VGG16 đã chứng tỏ là một cột mốc quan trọng trong nhiệm vụ của nhân loại là làm cho máy tính "nhìn thấy" thế giới. Rất nhiều nỗ lực đã được đưa vào để cải thiện khả năng này trong lĩnh vực Thị giác máy tính (CV) trong một số thập kỷ. VGG16 là một trong những đổi mới quan trọng đã mở đường cho một số đổi mới tiếp theo trong lĩnh vực này.

Đó là mô hình Mạng thần kinh chuyển đổi (CNN) được đề xuất bởi Karen Simonyan và Andrew Zisserman tại Đại học Oxford. Ý tưởng về mô hình đã được đề xuất vào năm 2013, nhưng mô hình thực tế đã được gửi trong Thử thách ImageNet ILSVRC vào năm 2014. Thử thách nhận dạng hình ảnh quy mô lớn ImageNet (ILSVRC) là một cuộc thi hàng năm đánh giá các thuật toán để phân loại hình ảnh (và phát hiện đối tượng) tại một quy mô lớn. Họ đã làm tốt trong thử thách nhưng không thể giành chiến thắng.

2.2.1 Cấu hình:

Một chồng gồm nhiều lớp tích chập (thường là 1, 2 hoặc 3) của bộ lọc có kích thước 3 x 3, một bước và phần đệm 1, tiếp theo là lớp tổng hợp tối đa có kích thước 2 x 2, là khối xây dựng cơ bản cho tất cả các cấu hình này. Các cấu hình khác nhau của ngăn xếp này được lặp lại trong cấu hình mạng để đạt được các độ sâu khác nhau. Số được liên kết với mỗi cấu hình là số lớp có tham số trọng lượng trong đó.

Các ngăn xếp tích chập được theo sau bởi ba lớp được kết nối đầy đủ, hai lớp có kích thước 4.096 và lớp cuối cùng có kích thước 1.000. Lớp cuối cùng là lớp đầu ra có kích hoạt Softmax. Kích thước 1.000 đề cập đến tổng số lớp có thể có trong ImageNet.

VGG16 đề cập đến cấu hình "D" trong bảng liệt kê bên dưới. Cấu hình "C" cũng có 16 lớp trọng lượng. Tuy nhiên, nó sử dụng bộ lọc 1 x 1 làm lớp tích chập cuối cùng trong ngăn xếp 3, 4 và 5. Lớp này được sử dụng để tăng tính phi tuyến tính của các hàm quyết định mà không ảnh hưởng đến trường tiếp nhận của lớp.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Hình 2.1: Các cấu hình khác nhau của VGG

2.2.2 Kiến trúc:

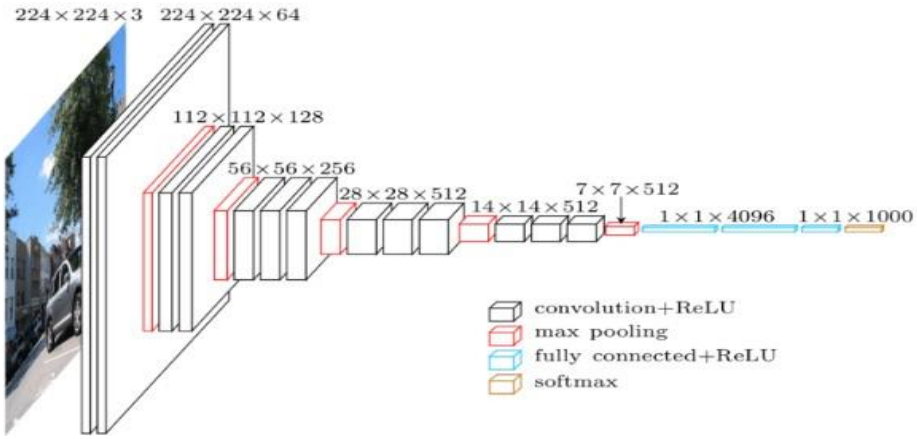
Đầu vào cho bất kỳ cấu hình mạng nào được coi là hình ảnh có kích thước cố định 224 x 224 với ba kênh – R, G và B. Quá trình tiền xử lý duy nhất được thực hiện là chuẩn hóa các giá trị RGB cho mỗi pixel. Điều này đạt được bằng cách trừ đi giá trị trung bình từ mỗi pixel.

Hình ảnh được truyền qua ngăn xếp đầu tiên gồm 2 lớp tích chập có kích thước tiếp nhận rất nhỏ là 3 x 3, tiếp theo là kích hoạt ReLU. Mỗi lớp trong số hai lớp này chứa 64 bộ lọc. Bước tích chập được cố định ở 1 pixel và phần đệm là 1 pixel. Cấu hình này duy trì độ phân giải không gian và kích thước của bản đồ kích hoạt đầu ra giống như kích thước hình ảnh đầu vào. Sau đó, các bản đồ kích hoạt được chuyển qua

tổng hợp tối đa không gian trên cửa sổ 2 x 2 pixel, với bước tiến là 2 pixel. Điều này làm giảm một nửa kích thước của các kích hoạt. Do đó, kích thước của các kích hoạt ở cuối ngăn xếp đầu tiên là 112 x 112 x 64.

Sau đó, các kích hoạt sẽ chảy qua ngăn xếp thứ hai tương tự, nhưng với 128 bộ lọc so với 64 trong ngăn xếp đầu tiên. Do đó, kích thước sau ngăn xếp thứ hai trở thành 56 x 56 x 128. Tiếp theo là ngăn xếp thứ ba với ba lớp tích chập và một lớp nhóm tối đa. Không. trong số các bộ lọc được áp dụng ở đây là 256, làm cho kích thước đầu ra của ngăn xếp là 28 x 28 x 256. Tiếp theo là hai ngăn xếp gồm ba lớp tích chập, mỗi ngăn chứa 512 bộ lọc. Đầu ra ở cuối cả hai ngăn xếp này sẽ là 7 x 7 x 512.

Theo sau các ngăn xếp của các lớp tích chập là ba lớp được kết nối đầy đủ với một lớp làm phẳng ở giữa. Hai lớp đầu tiên có 4.096 nơ-ron mỗi lớp và lớp được kết nối đầy đủ cuối cùng đóng vai trò là lớp đầu ra và có 1.000 nơ-ron tương ứng với 1.000 lớp có thể có cho bộ dữ liệu ImageNet. Lớp đầu ra được theo sau bởi lớp kích hoạt Softmax được sử dụng để phân loại theo danh mục.

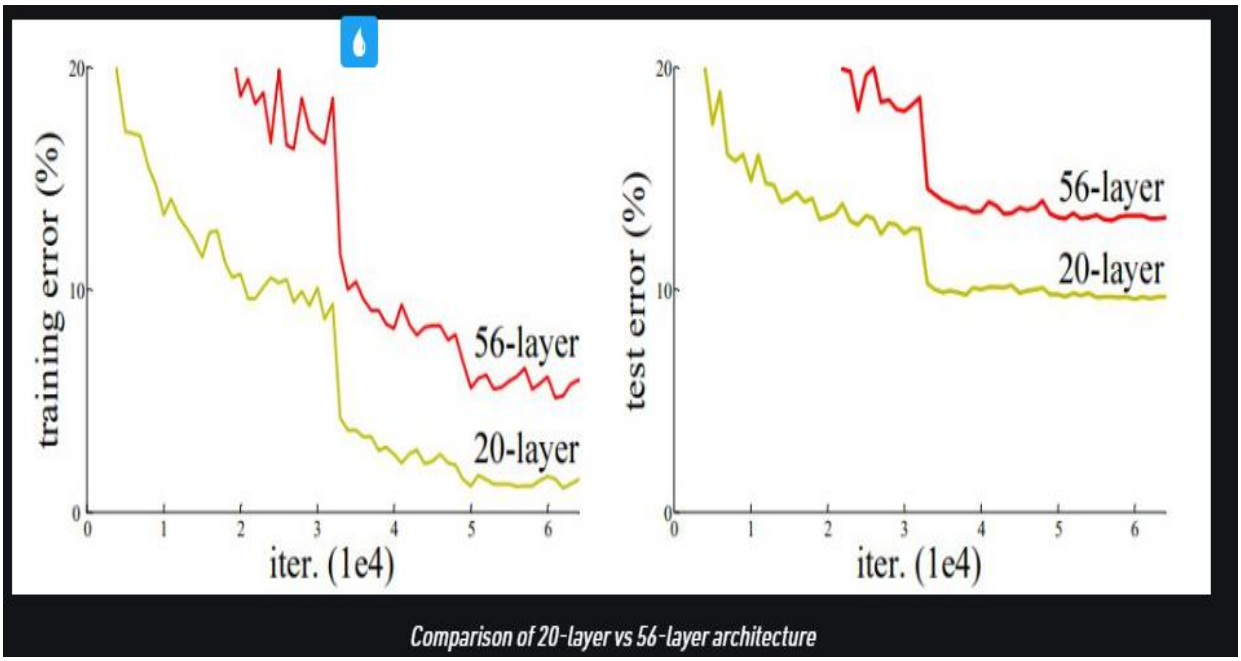


Hình 3.2: Kiến trúc mạng VGG16

2.3 Resnet-50

Sau khi kiến trúc dựa trên CNN đầu tiên (AlexNet) giành chiến thắng trong cuộc thi ImageNet 2012, mọi kiến trúc giành chiến thắng tiếp theo đều sử dụng nhiều lớp hơn trong mạng thần kinh sâu để giảm tỷ lệ lỗi. Điều này hoạt động với ít lớp hơn, nhưng khi chúng tôi tăng số lượng lớp, có một vấn đề phổ biến trong quá trình học

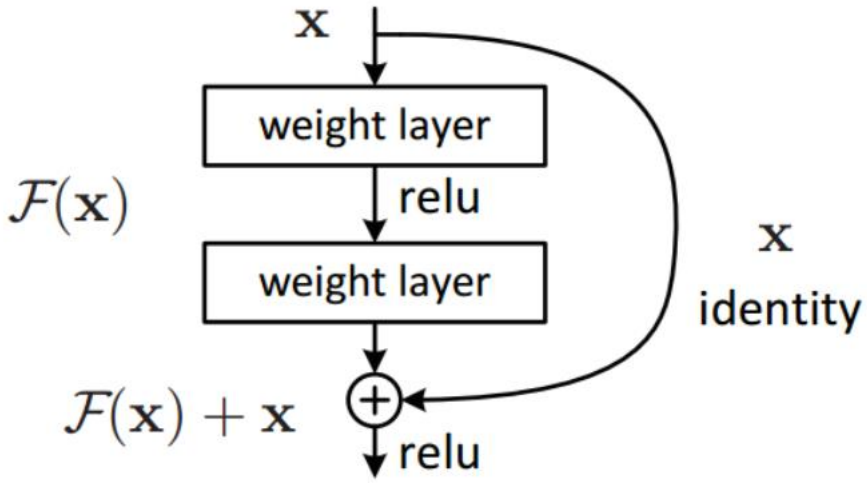
sâu liên quan đến độ dốc Vanishing/Exploding. Điều này làm cho độ dốc trở thành 0 hoặc quá lớn. Do đó, khi chúng ta tăng số lượng lớp, tỷ lệ lỗi huấn luyện và kiểm tra



Comparison of 20-layer vs 56-layer architecture

Trong biểu đồ trên, chúng ta có thể quan sát thấy rằng CNN 56 lớp cho tỷ lệ lỗi cao hơn trên cả tập dữ liệu huấn luyện và thử nghiệm so với kiến trúc CNN 20 lớp. Sau khi phân tích nhiều tỷ lệ lỗi hơn, các tác giả đã có thể đi đến kết luận rằng nguyên nhân là do vanishing/exploding gradient.(gradient biến mất/bùng nổ). ResNet, được đề xuất vào năm 2015 bởi các nhà nghiên cứu tại Microsoft Research đã giới thiệu một kiến trúc mới có tên là Mạng dư.(Residual Network)

ResNet gần như tương tự với các mạng nơ-ron tích chập cơ bản gồm có convolution, pooling, activation và fully-connected layer. Những kiến trúc trước đây thường cải tiến độ chính xác nhờ gia tăng chiều sâu của mạng CNN. Nhưng thực nghiệm cho thấy đến một ngưỡng độ sâu nào đó thì độ chính xác của mô hình sẽ bão hòa và thậm chí phản tác dụng và làm cho mô hình kém chính xác hơn. Ý tưởng chính của Resnet là sử dụng kết nối tắt. Các kết nối tắt (skip connection) giúp giữ thông tin không bị mất bằng cách kết nối từ layer sớm trước đó tới layer phía sau và bỏ qua một vài layers trung gian.



Ảnh bên trên hiển thị khối dư được sử dụng trong mạng. Xuất hiện một mũi tên cong xuất phát từ đầu và kết thúc tại cuối khối dư. Hay nói cách khác là sẽ bổ sung Input X vào đầu ra của layer, hay chính là phép cộng mà ta thấy trong hình minh họa, việc này sẽ chống lại việc đạo hàm bằng 0, do vẫn còn cộng thêm X. Với H(x) là giá trị dự đoán, F(x) là giá trị thật (nhãn), chúng ta muốn H(x) bằng hoặc xấp xỉ F(x). Việc F(x) có được từ x như sau:

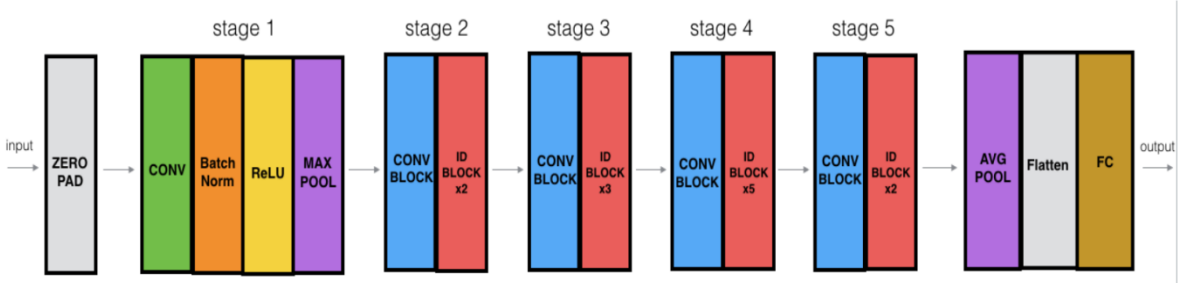
X->weight1-> ReLU -> weight2

Giá trị H(x) có được bằng cách:

F(x) + x -> ReLUF(x) + x -> ReLU

2.3.1 Kiến Trúc

Sơ đồ kiến trúc Resnet-50 mặc định của chúng tôi được hiển thị trong hình bên dưới:



"ID BLOCK" trong hình trên là viết tắt của từ Identity block và ID BLOCK x3 nghĩa là có 3 khối Identity block chồng lên nhau.

Nội dung hình trên như sau :

Zero-padding : Input với (3,3)

Stage 1 : Tích chập (Conv1) với 64 filters với shape(7,7), sử dụng stride (2,2). BatchNorm, MaxPooling (3,3).

Stage 2 : Convolutiontal block sử dụng 3 filter với size 64x64x256, f=3, s=1. Có 2 Identity blocks với filter size 64x64x256, f=3.

Stage 3 : Convolutional sử dụng 3 filter size 128x128x512, f=3,s=2. Có 3 Identity blocks với filter size 128x128x512, f=3.

Stage 4 : Convolutional sử dụng 3 filter size 256x256x1024, f=3,s=2. Có 5 Identity blocks với filter size 256x256x1024, f=3.

Stage 5 :Convolutional sử dụng 3 filter size 512x512x2048, f=3,s=2. Có 2 Identity blocks với filter size 512x512x2048, f=3.

The 2D Average Pooling : sử dụng với kích thước (2,2).

Fully Connected (Dense) : sử dụng softmax activation.

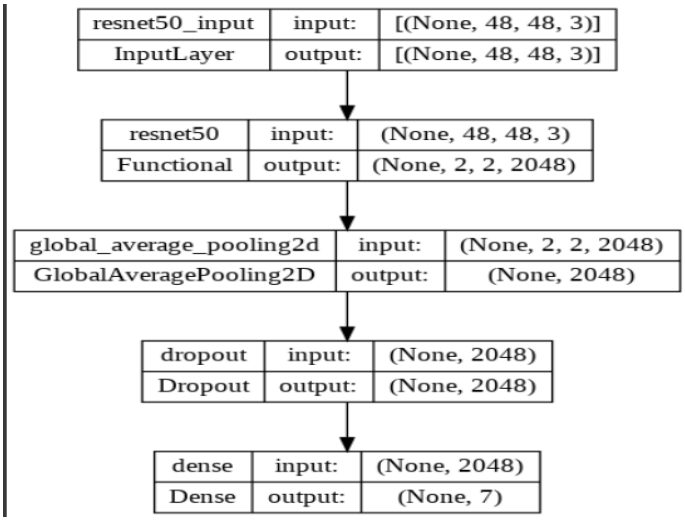
Chúng ta thực hiện thay đổi đầu vào trở thành 96x96x1 và thay đổi đầu ra trở thành:

GlovalAveragePoooling2D(): Tổng hợp toàn cầu ngưng tụ tất cả các bản đồ tính năng thành một bản đồ duy nhất, gộp tất cả các thông tin liên quan vào một bản đồ có thể dễ dàng hiểu được bằng một lớp phân loại dày đặc thay vì nhiều lớp

Dropout(): Trong mạng Neural Network, kỹ thuật Dropout là việc chúng ta sẽ bỏ qua một vài unit trong suốt quá trình train trong mô hình, những unit bị bỏ qua được lựa chọn ngẫu nhiên. Ở đây, chúng ta hiểu "bỏ qua - ignoring" là unit đó sẽ không tham gia và đóng góp vào quá trình huấn luyện (lan truyền tiến và lan truyền ngược).Về chức năng là để chống over-fitting

Fully Connected(Dense): Sử dụng Sofmax activation

Sơ đồ kiến trúc Resnet-50 sẽ sử dụng của chúng tôi được hiển thị trong hình bên dưới



Giải pháp ResNet đưa ra đơn giản hơn và tập trung vào cải thiện thông tin thông qua độ dốc của mạng. Sau ResNet hàng loạt biến thể của kiến trúc này được giới thiệu. Thực nghiệm cho thấy những kiến trúc này có thể được huấn luyện mạng nơ-ron với độ sâu hàng nghìn lớp và nó nhanh chóng trở thành kiến trúc phổ biến nhất trong thị giác máy tính

2.4 Viola-Jones

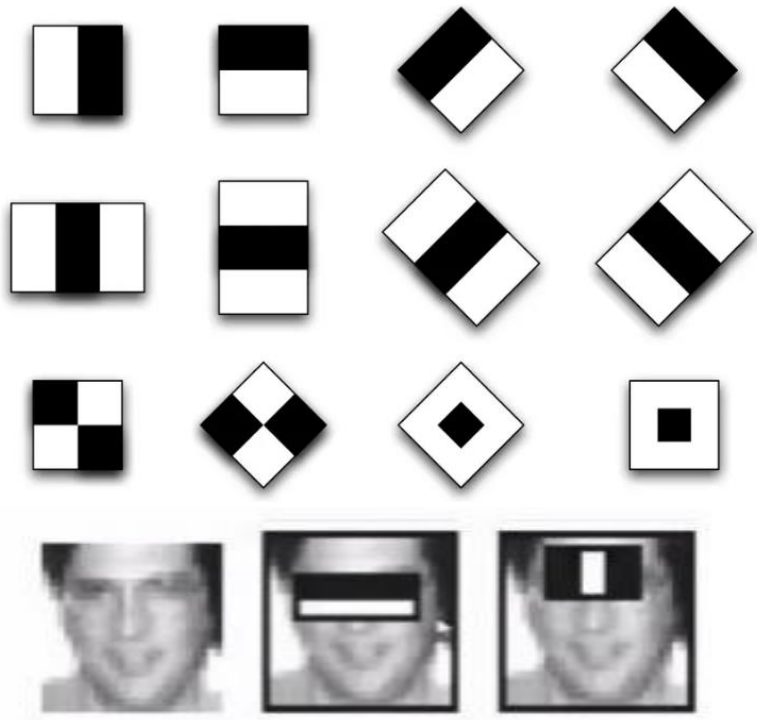
1. Thuật toán Viola-Jones

1.1. Các đặc trưng Haar-Like

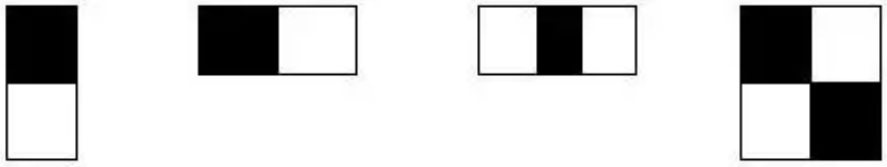
Các đặc trưng Haar-Like là những hình chữ nhật được phân thành các vùng khác nhau.

Ý tưởng : độ sáng tối của các vùng trên gương mặt là khác nhau. Ví dụ: vùng mắt tối hơn vùng má, vùng mũi sáng hơn vùng hai bên

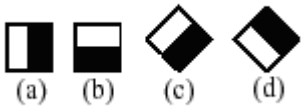
Kết quả của mỗi đặc trưng được tính bằng hiệu của tổng các pixel trong miền ô trắng trừ đi tổng các pixel trong miền ô đen.



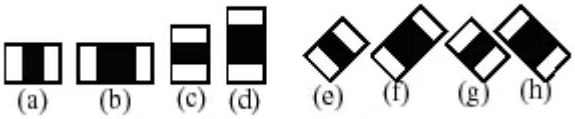
Đặc trưng do Viola và Jones công bố gồm 4 đặc trưng cơ bản để xác định khuôn mặt người. Mỗi đặc trưng Haar-Like là sự kết hợp của hai hay ba hình chữ nhật trắng hay đen như trong hình sau:



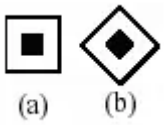
a. Đặc trưng cạnh(edge feature):



b. Đặc trưng đường(line feature):

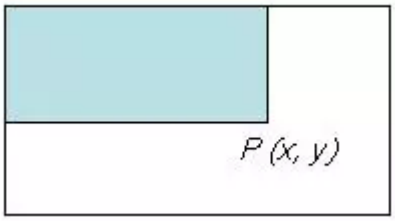


c. Đặc trưng xung quanh tâm(center-surround features)



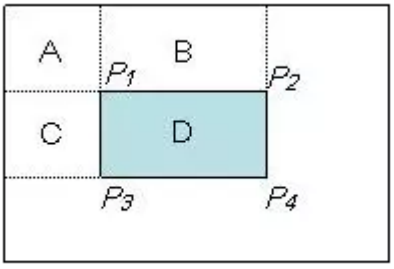
1.2. Integral Image

Viola và Joines đưa ra một khái niệm gọi là Integral Image, là một mảng 2 chiều với kích thước bằng với kích thước của ảnh cần tính đặc trưng Haar-Like, với mỗi phần tử của mảng này được tính bằng cách tính tổng của điểm ảnh phía trên (dòng-1) và bên trái (cột-1) của nó.



Công thức tính: $P(x,y) = \sum_{x' \le x, y' \le y} i(x', y')$

Sau khi tính được Integral Image, việc tính tổng các giá trị mức xám của một vùng bất kỳ nào đó trên ảnh thực hiện rất đơn giản theo cách sau:



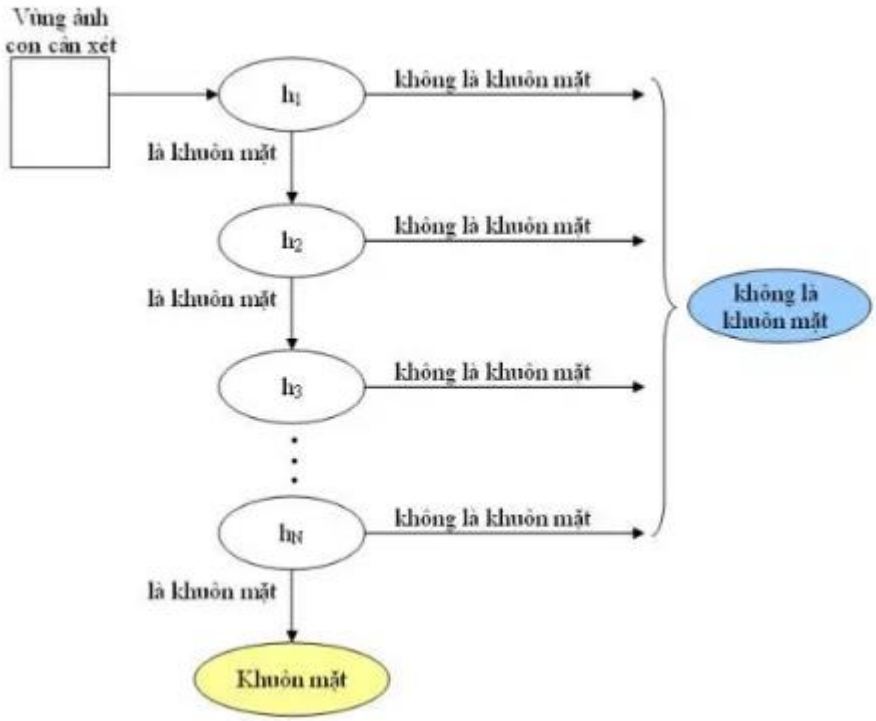
D = A + B + C + D – (A+B) – (A+C) + A

Với A + B + C + D chính là giá trị tại điểm P4 trên Integral Image, tương tự như vậy A+B là giá trị tại điểm P2, A+C là giá trị tại điểm P3, và A là giá trị tại điểm P1.

1.3. Adaboost

AdaBoost là một bộ phân loại mạnh phi tuyến phức dựa trên hướng tiếp cận boosting được Freund và Schapire đưa ra vào năm 1995. Adaboost cũng hoạt động trên nguyên tắc kết hợp tuyến tính các bộ phân loại yếu để hình thành một bộ phân loại mạnh

Viola và Jones dùng AdaBoost kết hợp các bộ phân loại yếu sử dụng các đặc trưng Haar-like theo mô hình phân tầng (cascade) như sau:



Trong đó, h(k) là các bộ phân loại yếu, được biểu diễn như sau:

$$h(k) = \begin{cases} 1 & nếu\ p_k f_k(x) < p_k \theta_k \\ 0 & Ngược\ lại \end{cases}$$

x: cửa sổ con cần xét

θk : ngưỡng

f_k: giá trị của đặc trưng Haar-like

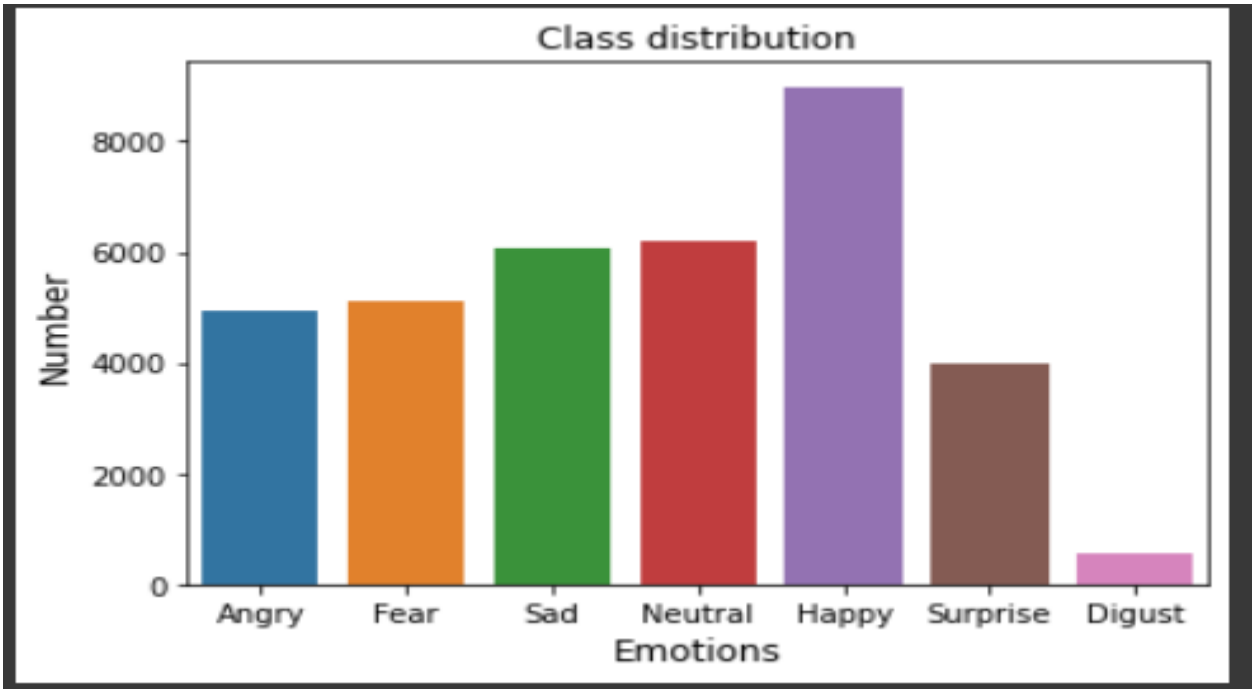
p_k: hệ số quyết định chiều của phương trình

2. Bộ Dữ Liệu

Trong dự án này, chúng tôi đã sử dụng bộ dữ liệu do trang web Kaggle cung cấp, bao gồm khoảng 37.000 hình ảnh khuôn mặt có tỷ lệ xám 48 × 48 pixel được cấu trúc tốt. Các hình ảnh được xử lý theo cách sao cho các khuôn mặt gần như được căn giữa và mỗi mặt chiếm khoảng cùng một lượng không gian trong mỗi hình ảnh. Mỗi hình ảnh có được xếp vào một trong bảy loại thể hiện những cảm xúc khác nhau trên khuôn mặt. Những cảm xúc trên khuôn mặt này đã được phân loại thành: 0=Giận dữ, 1=Ghê tởm, 2=Sợ hãi, 3=Vui, 4=Buồn, 5=Bất ngờ và 6=Bình thường. Hình 1 mô tả một ví dụ cho mỗi loại nét mặt. Ngoài số lớp ảnh (a số từ 0 đến 6), các hình ảnh đã cho được chia thành ba bộ khác nhau là đào tạo, xác nhận, và bộ kiểm tra. Có khoảng 29.000 hình ảnh đào tạo, 4.000 hình ảnh xác thực và 4.000 hình ảnh để thử nghiệm. Sau khi đọc dữ liệu pixel thô, chúng tôi đã chuẩn hóa chúng bằng cách trừ đi giá trị trung bình của các hình ảnh đào tạo từ mỗi hình ảnh bao gồm cả những hình ảnh trong quá trình xác thực và kiểm tra bộ. Với mục đích tăng cường dữ liệu, chúng tôi đã tạo ra các hình ảnh phản chiếu bằng cách lật các hình ảnh trong tập huấn luyện theo chiều ngang.

Figure 1: Examples of seven facial emotions that we consider in this classification problem. (a) angry, (b) neutral, (c) sad, (d) happy, (e) surprise, (f) fear, (g) disgust

Thống kê bộ dữ liệu được hiện thi ở hình bên dưới:



Phần 4: THỰC NGHIỆM

Trong phần này, chúng tôi sẽ chứng minh cách chúng tôi tiến hành các thí nghiệm trên phương án cơ sở, mô hình VGG-16 và mô hình Resnet-50 một cách chi tiết.

Để đánh giá các mạng của chúng tôi, chúng tôi so sánh tổn thất sau một số lượng kỷ nguyên nhất định (50). Các đường cong của tổn thất giảm được thể hiện như sau:

Các kết quả thí nghiệm bao gồm hai phần. Đầu tiên, chúng tôi tập trung vào độ chính xác của phát hiện bằng cách so sánh tổn thất đào tạo, xác nhận và kiểm tra giữa tất cả mô hình được minh họa ở trên. Tiếp theo chúng tôi sẽ thực hiện so sánh của hiệu suất của mô hình trên bộ kiểm định.

4.1 PCA và Học Máy SVM

Sử dụng PCA với n_components = 130.

```
n_components = 130
pca = RandomizedPCA(n_components=n_components, whiten=True).fit(X_train)
sum(pca.explained_variance_ratio_)
```

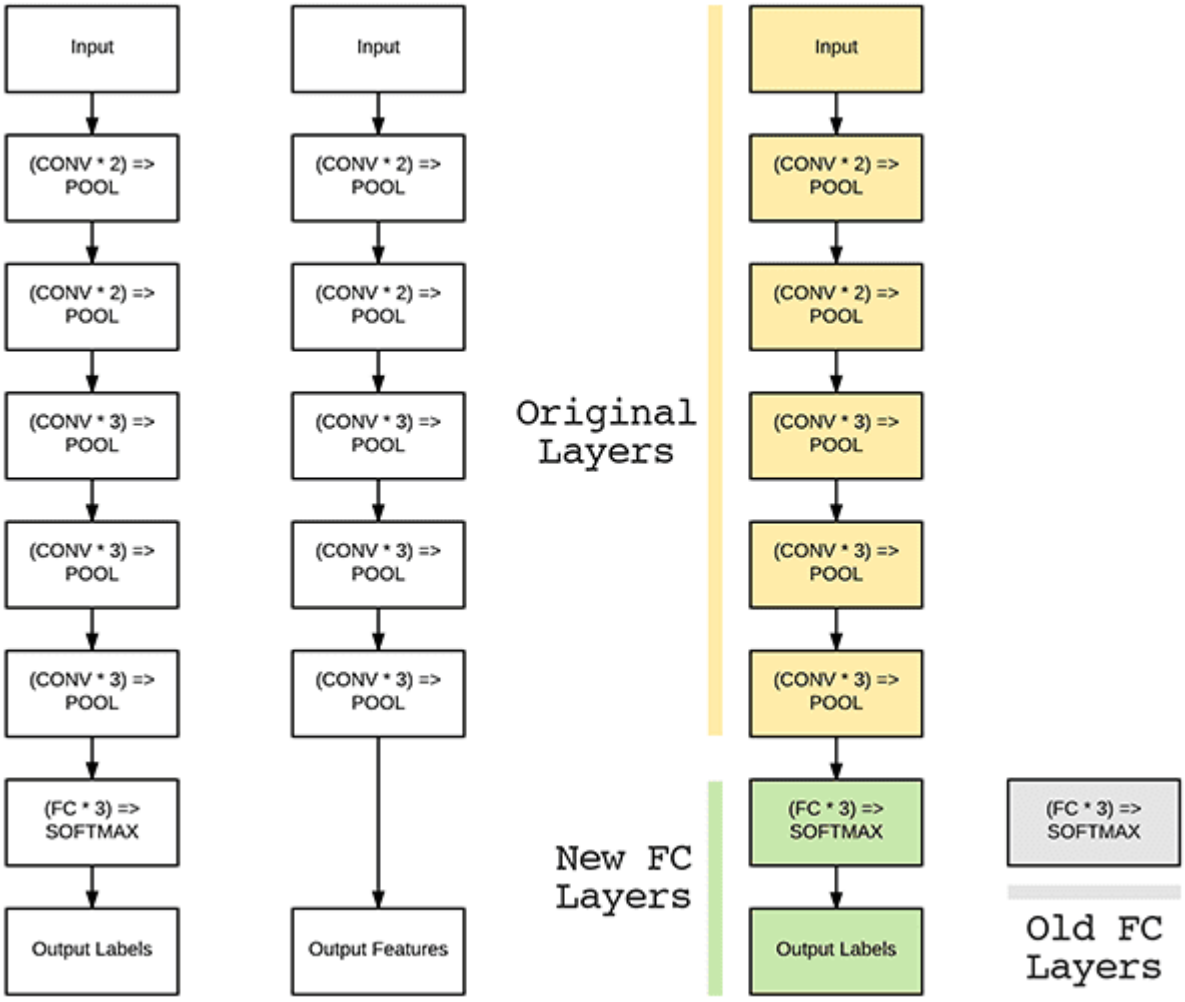
```
0.9141887287551071
```

Sau đó dùng GridSearchCV để tìm các siêu tham số C và gamma.

```
param_grid = {
    'C': [0.01, 0.05, 0.1, 0.5, 1, 5, 1e1, 5e1, 1e2, 5e2, 1e3, 5e3, 1e4, 5e4, 1e5, 5e5, 1e6],
    'gamma': [1e-8, 5e-8, 1e-7, 5e-7, 1e-6, 5e-6, 1e-5, 5e-5, 1e-4, 5e-4, 1e-3, 5e-3, 1e-2, 5e-2],
}
clf = GridSearchCV(SVC(kernel='sigmoid', class_weight='balanced'),
param_grid)
clf = clf.fit(X_train_pca, y_train)
```

4.2 VGG-16

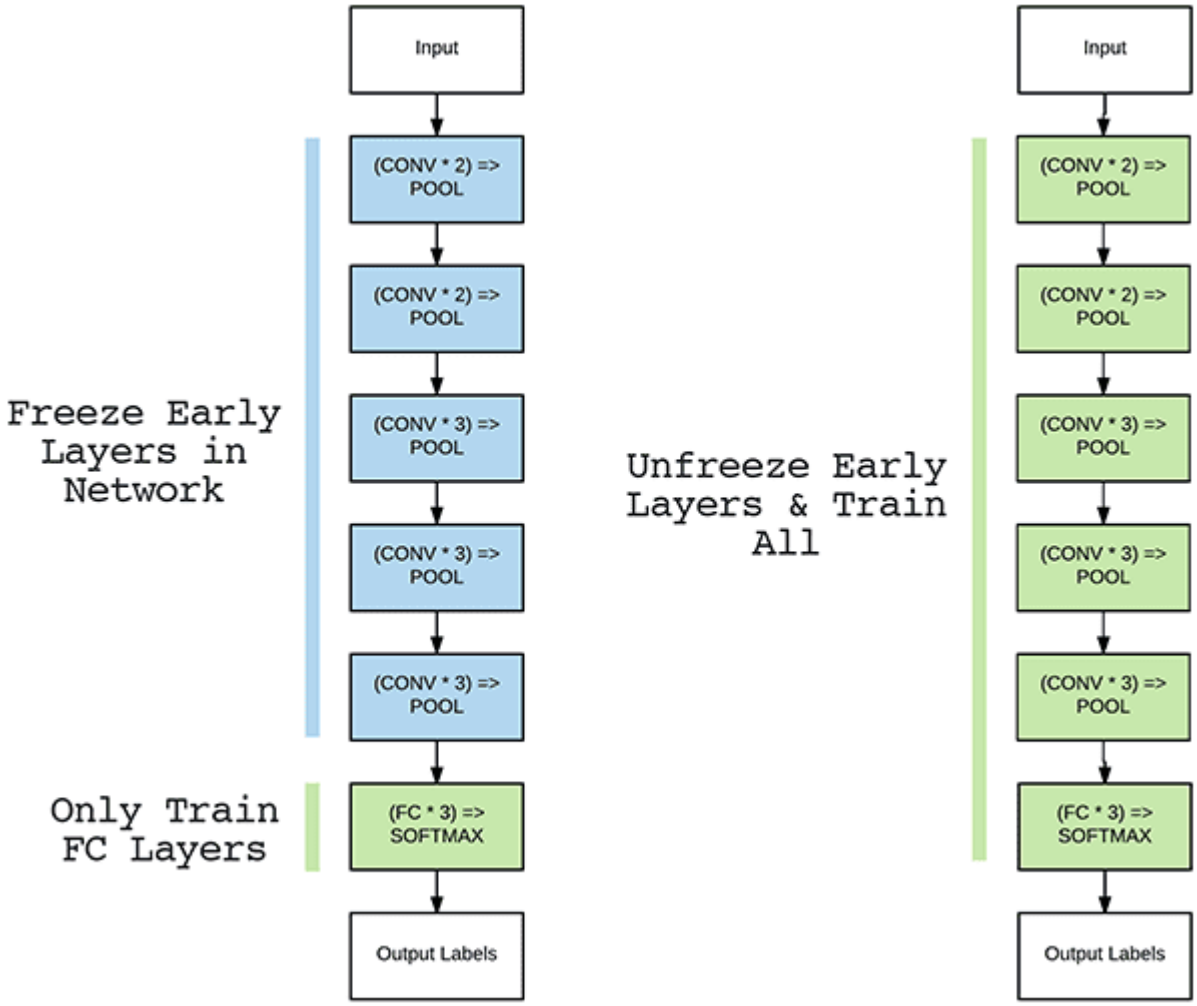
Bước 1: Vì bộ dữ liệu chỉ có 7 lớp trong khi đầu ra của model VGG16 pretrained gồm 1000 lớp nên tiên hành loại bỏ các Lớp FC ban đầu và thay thế chúng bằng một đầu FC hoàn toàn mới phù hợp với bộ dữ liệu.



Hình 4.1: Mạng VGG16 ban đầu và sau khi được thay đổi

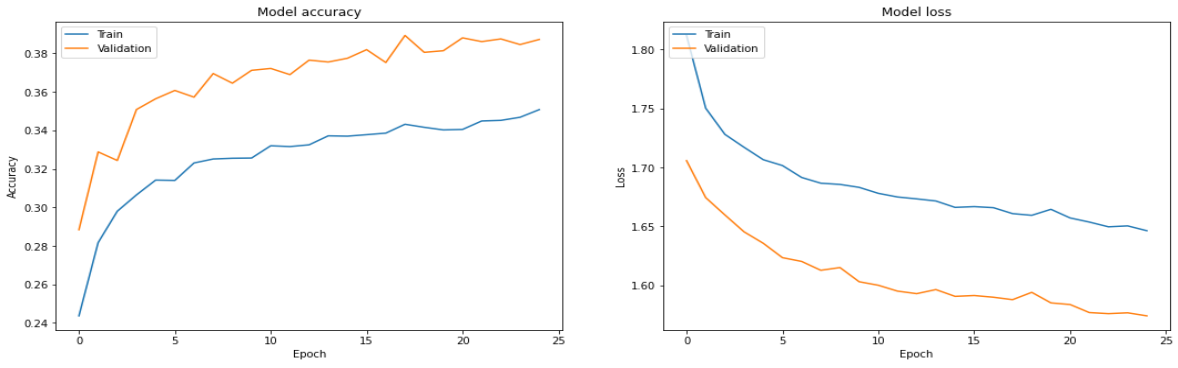
Bước 2: Đóng băng các lớp CONV chỉ cho phép gradient lan truyền ngược qua các lớp FC. Các lớp CONV đã học được các bộ lọc phân biệt, phong phú trong khi các lớp FC là hoàn toàn mới và hoàn toàn ngẫu nhiên.

Nếu cho phép gradient lan truyền ngược từ các giá trị ngẫu nhiên này trong toàn bộ mạng, có nguy cơ phá hủy các tính năng mạnh mẽ này.



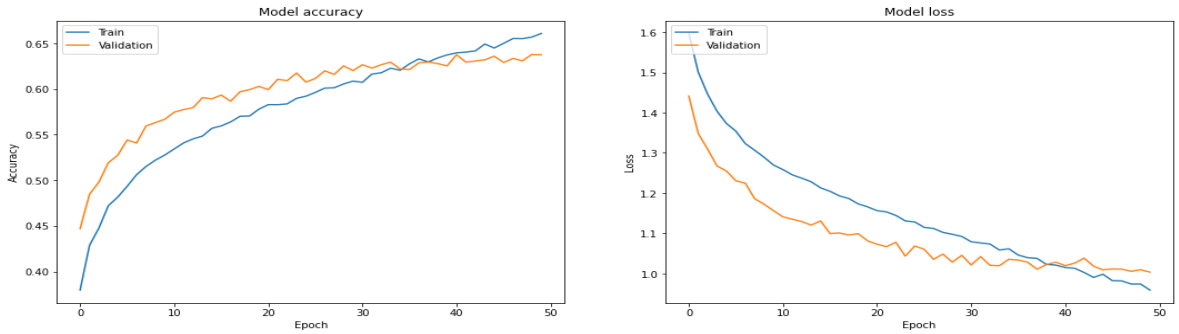
Hình 4.2: Đóng băng và mở băng mạng neuron

Bước 3: Sau đó mở băng toàn bộ mạng rồi tiếp tục huấn luyện.



Hình 4.3: Loss và accuracy sau khi train 50 epoch

Sau khi đóng băng các lớp CONV train 50 epoch thấy loss của tập train giảm mạnh trong khi tập validation giảm chậm và có xu hướng tăng nên dừng lại.

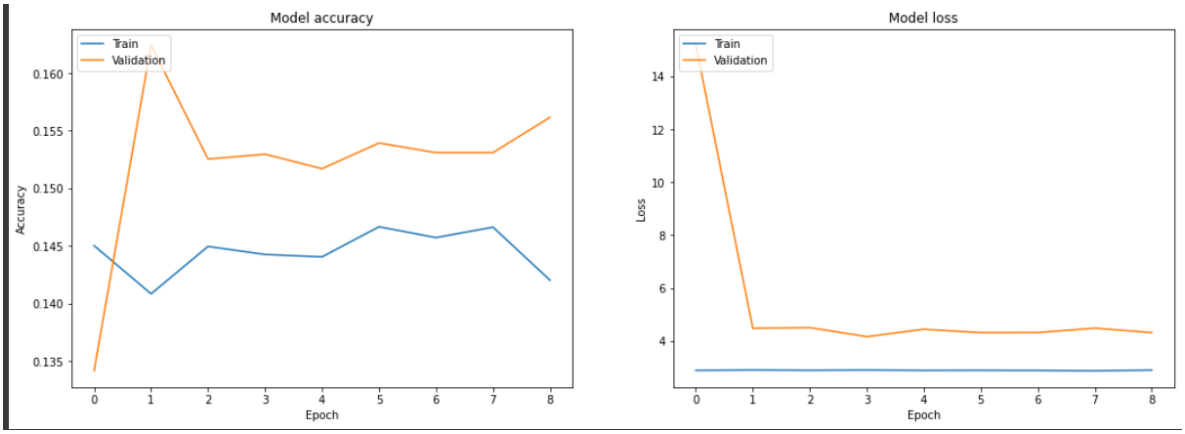


Hình 4.4: Loss và accuracy sau khi train 50 epoch tiếp theo

Mở băng các lớp CONV và tiếp tục train 50 epoch thì model bị overfit.

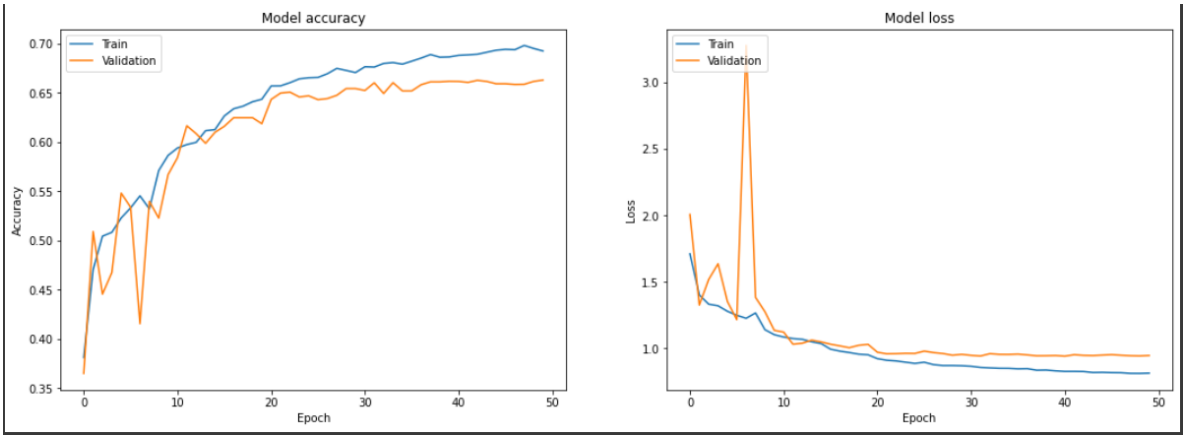
4.3 Resnet-50

Đối với mô hình Resnet-50 mà chúng ta sử dụng lại trọng số đào tạo được huấn luyện trước đây được thể hiện như sau:



Ở phần thông số mô hình chúng ta có chỉ số early stopping nếu tỷ lệ trên bộ kiểm định của mô hình không có sự tăng lên sau 7 epochs (patience = 7) thì mô hình sẽ ngưng đào tạo. Sau khi đến epochs thứ 8 thì mô hình đã dừng lại và chúng ta có thể thấy bài toán chúng ta thực hiện không phù hợp với trọng số mô hình đã đào tạo trước. Nên chúng ta sẽ tiến hành đào tạo lại từ đầu

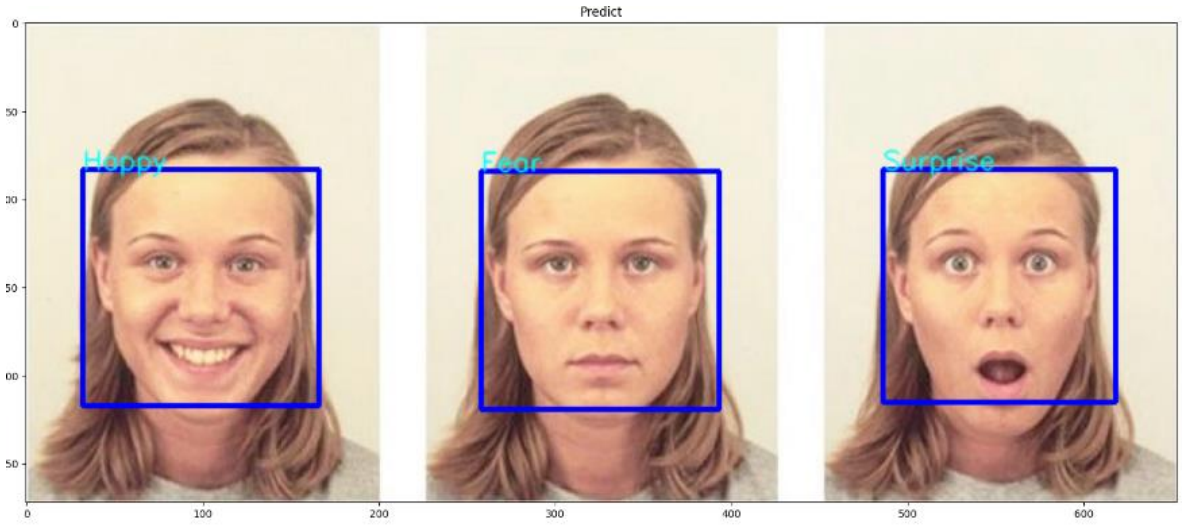
Đối với mô hình Resnet-50 cho thực hiện đào tạo lại từ đầu, kết quả tổn thất và hiệu suất mô hình được thể hiện như sau



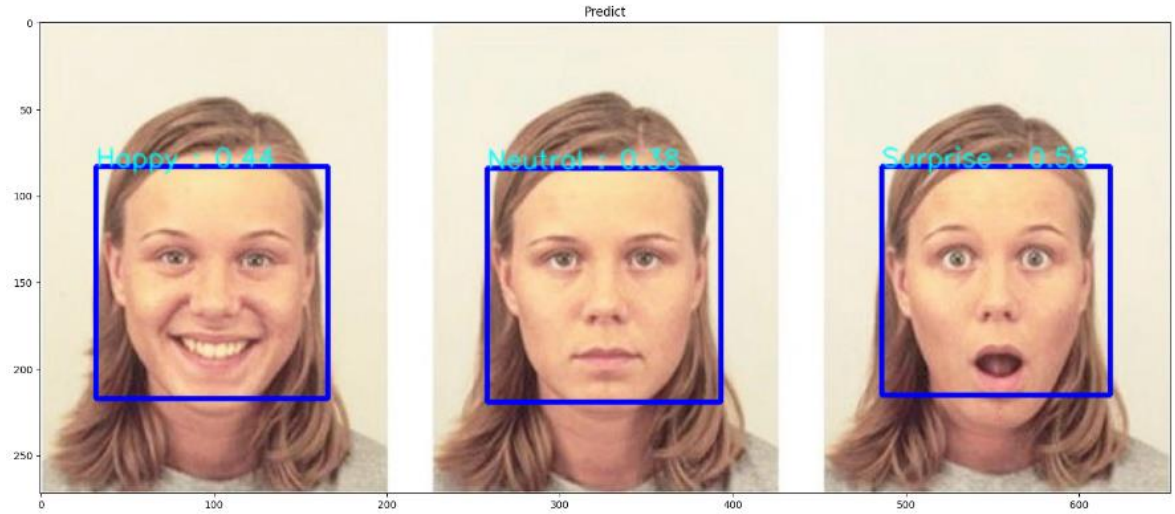
Chúng ta có thể thấy sự tăng trưởng ổn định về hiệu suất và kết quả tổn thất giảm dần về 0, mặc dù có sự bùng nổ về độ dốc ở khoảng epochs 15 - 25 nhưng nhìn chung thì tổn thất ở các epochs sau đã duy trì ổn định và có xu hướng giảm,

4.2 Kết Quả

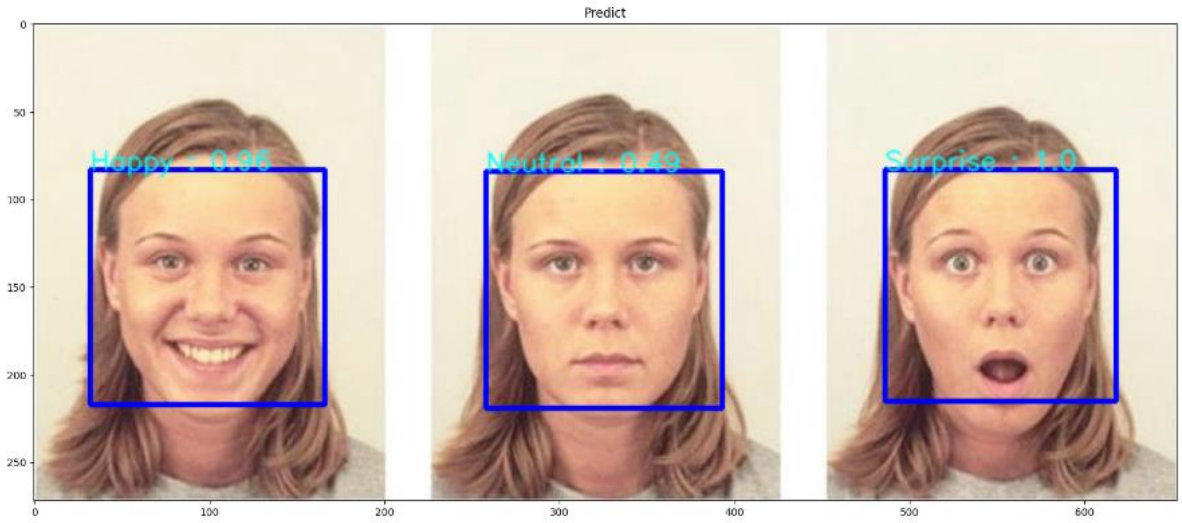
	SVM	**VGG16**	**Resnet-50**
Accuracy	0.5507	0.6412	**0.66186**



Hình 4.1: Kết quả thực nghiệm bằng SVM



Hình 4.2: Kết quả thực nghiệm bằng VGG16



Hình 4.3: Kết quả thực nghiệm bằng Resnet-50

Nhận xét:

Kết quả thử nghiệm thực tế cho thấy 2 mô hình khá nhạy khi nhận biết cảm xúc Happy, khá kém với cảm xúc Disgust. Điều này khá dễ hiểu khi số lượng của bộ dữ liệu Happy chiếm hơn 25% trong bộ 7 dữ liệu cảm xúc trong khi bộ dữ liệu Disgust chiếm số lượng ít nhất

Fear với Angry, Neutral với Sad có biểu cảm khá giống nhau nên model thường nhầm lẫn. Kết quả là cho ra độ chính xác không cao.

Model SVM còn bị nhầm lẫn giữa biểu cảm Neutral và Fear, model VGG16 cho kết quả đúng với các biểu cảm nhưng độ chính xác chưa cao, model Resnet-50 cho kết quả đúng với độ chính xác cao.

TÀI LIỆU THAM KHẢO

[1]: http://cs231n.stanford.edu/reports/2016/pdfs/005_Report.pdf

[2]: https://viblo.asia/p/gioi-thieu-mang-resnet-vyDZOa7R5wj

[3]: https://www.mygreatlearning.com/blog/introduction-to-vgg16/

[4]: https://pyimagesearch.com/2019/06/03/fine-tuning-with-keras-and-deep-learning/

[5]: https://machinelearningcoban.com/2017/04/09/smv/

[6]: https://viblo.asia/p/gioi-thieu-ve-support-vector-machine-svm-6J3ZgPVElmB