

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



FACIAL EXPRESSION RECOGNITION

Giảng viên: Đỗ Văn Tiến

Lớp: CS338.M21

Sinh viên thực hiện:		
STT	Họ tên	MSSV
1	Nguyễn Thị Như Ý	19522555
2	Nguyễn Tấn Tú	19522454
3	Đinh Hoàng Linh Đan	19521309
4	Trần Nguyễn Quỳnh Anh	19521217

TP. HỒ CHÍ MINH – 6/2022

Mục Lục

1. GIỚI THIỆU	1
2. NỘI DUNG	1
2.1. Input, Output	1
2.2. Ứng dụng	2
2.3. Dataset	2
2.4. Model.....	3
2.4.1. Giới thiệu ResNet	3
2.4.2. Training Model.....	4
2.4.3. So sánh 3 model:	8
2.5. Hướng phát triển.....	8
2.6. Demo:	9
2.6.1. Xây dựng web API:	9
2.6.2. Hình ảnh Demo:	9
3. KẾT LUẬN	10
4. TÀI LIỆU THAM KHẢO.....	11
PHỤ LỤC PHÂN CÔNG NHIỆM VỤ	12

1. GIỚI THIỆU

Nhận diện biểu cảm biểu cảm là một trong những bản năng của con người

Con người thể hiện cảm xúc thông qua các biểu cảm khuôn mặt, luôn tồn tại một số đặc trưng chung trên khuôn mặt bất kể độ tuổi, vị trí địa lí hay điều kiện sống... Dựa vào đặc trưng này, ta rút trích ra các đặc điểm quan trọng của cảm xúc, mô hình hóa, và dạy cho máy tính hiểu được cảm xúc đó.

Ta chia biểu cảm khuôn mặt vào bảy loại sắc thái chính: tức giận(angry), sợ hãi(fear), ngạc nhiên (surprised), buồn(sad), chán ghét(disgusted), hạnh phúc(happy), trung lập(neutral).



Hình 1: biểu cảm gương mặt người

2. NỘI DUNG

2.1. Input, Output

- Input: Video webcam hoặc bức ảnh chứa mặt người
 - + Quay/ chụp chính diện
 - + Ánh sáng tốt
 - + Không vật cản lớn trên gương mặt

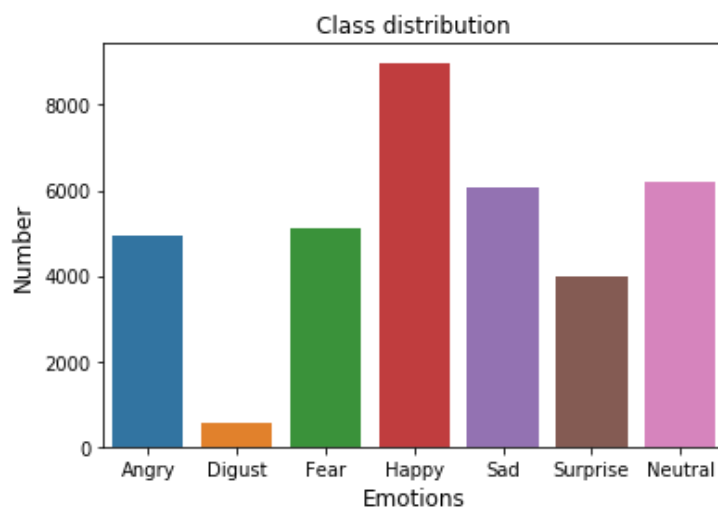
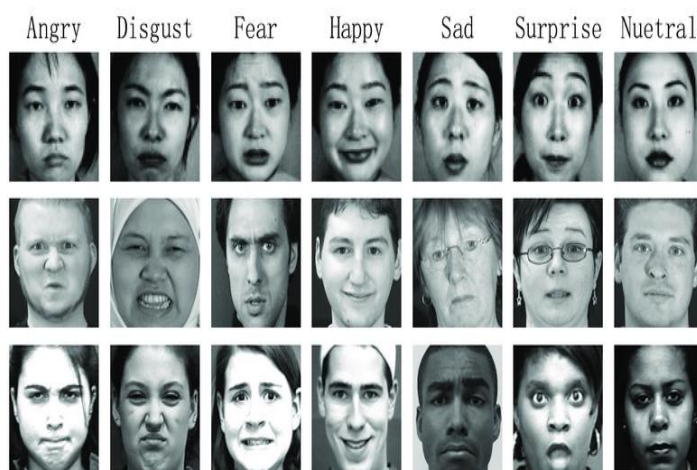
- Output: Dự đoán biểu cảm gương mặt có trong ảnh

2.2. Ứng dụng

- Trong lĩnh vực tiếp thị: để phân tích phản hồi của khách hàng
- Xác định những người có hành vi đáng ngờ trong đám đông từ đó có thể ngăn chặn tội phạm, khủng bố.
- Quản lý dịch vụ khách hàng : phân tích mức độ hài lòng của khách hàng.
- Sử dụng trong những ứng dụng game thực tế ảo

2.3. Dataset

- Dataset: Fer2013 gồm 35887 ảnh xám kích cỡ 48x48 chia thành 2 tập training và validation theo tỉ lệ lần lượt là 0.8 và 0.2
- Có 7 lớp biểu cảm:



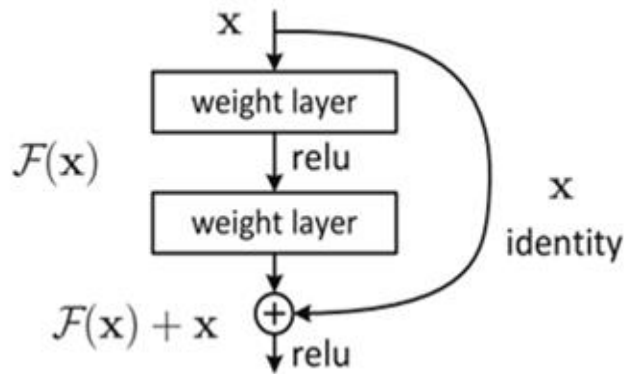
	emotion	number
0	Angry	4953
1	Disgust	547
2	Fear	5121
3	Happy	8989
4	Sad	6077
5	Surprise	4002
6	Neutral	6198

Hình 2: Dataset Fer2013

2.4. Model

2.4.1. Giới thiệu ResNet

Ý tưởng chính của ResNet là sử dụng kết nối tắt (Skip connection) đồng nhất để xuyên qua một hay nhiều lớp. Các kết nối tắt 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.



Hình 3: Model Resnet

ResNet gần như tương tự với các mạng gồm có convolution, pooling, activation và fully-connected layer.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Hình 4: Các model Resnet

2.4.2. Training Model

Nhóm sử dụng 3 model ResNet : ResNet-50, ResNet-101 và ResNet-15

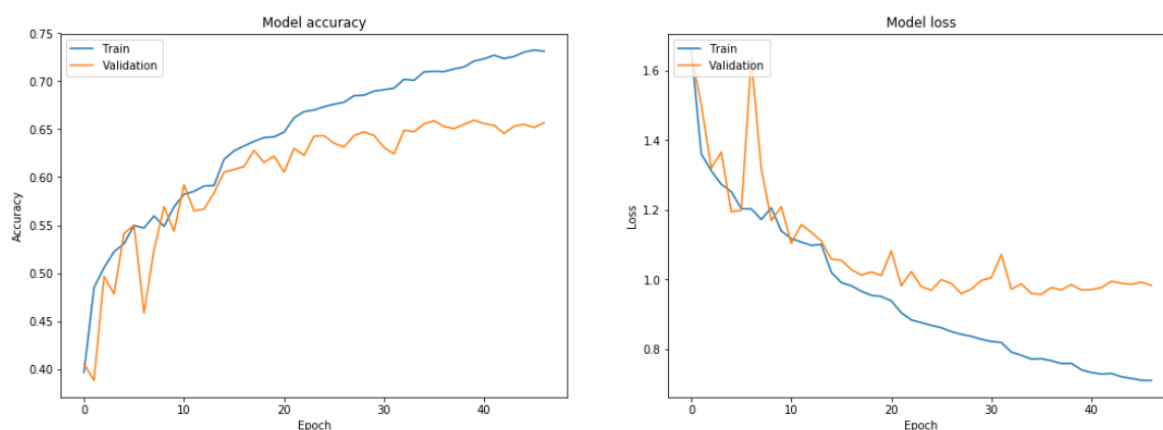
ResNet50

```
model = Sequential()

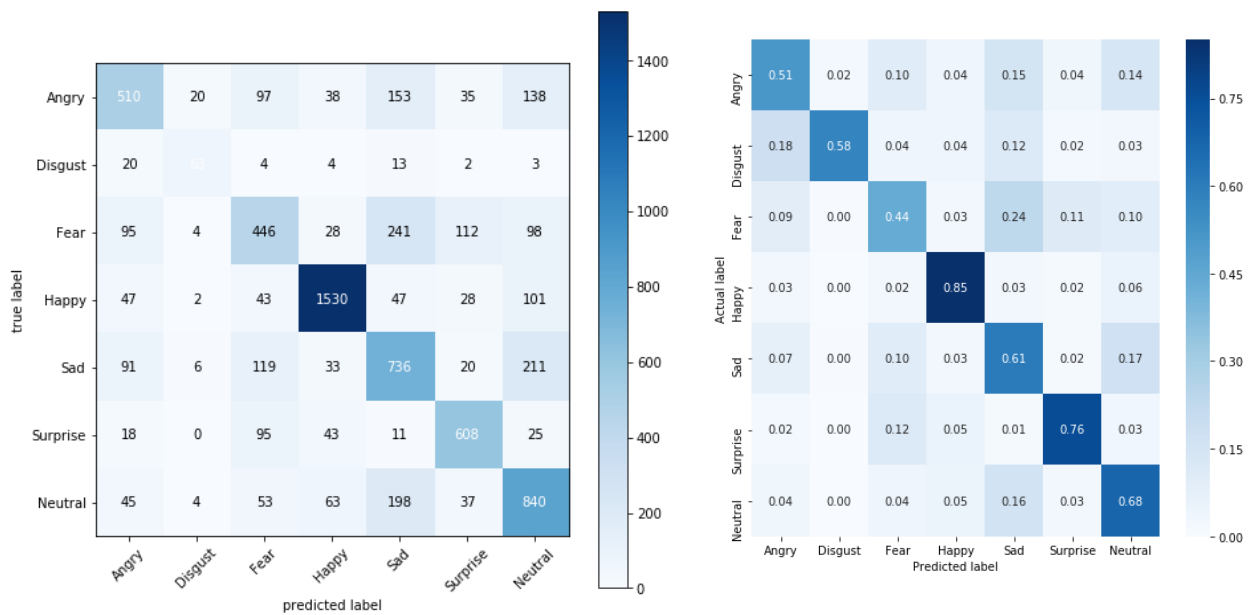
pretrained_model = tf.keras.applications.resnet50.ResNet50(include_top=False,
                                                            input_shape=(IMG_WIDTH, IMG_HEIGHT, 3), classes=7,
                                                            weights='../input/resnet50/resnet50_weights_tf_dim_
                                                            ordering_tf_kernels_notop.h5')
# for layer in pretrained_model.layers:
#     layer.trainable=False

model.add(pretrained_model)
# model.add(Flatten())
model.add(GlobalAveragePooling2D())
model.add(Dropout(0.2))
# Output layer
model.add(Dense(7, activation='softmax'))
# Compile the model
model.compile(optimizer=Adam(learning_rate=0.0005), loss=tf.keras.losses.SparseCategoricalCr
ossentropy(), metrics=['accuracy'])
model.summary()
```

Hình 5: Model ResNet-50



Hình 6: Accuracy và Loss của Mode ResNet50



Hình 7: Cofusion matrix của model ResNet50

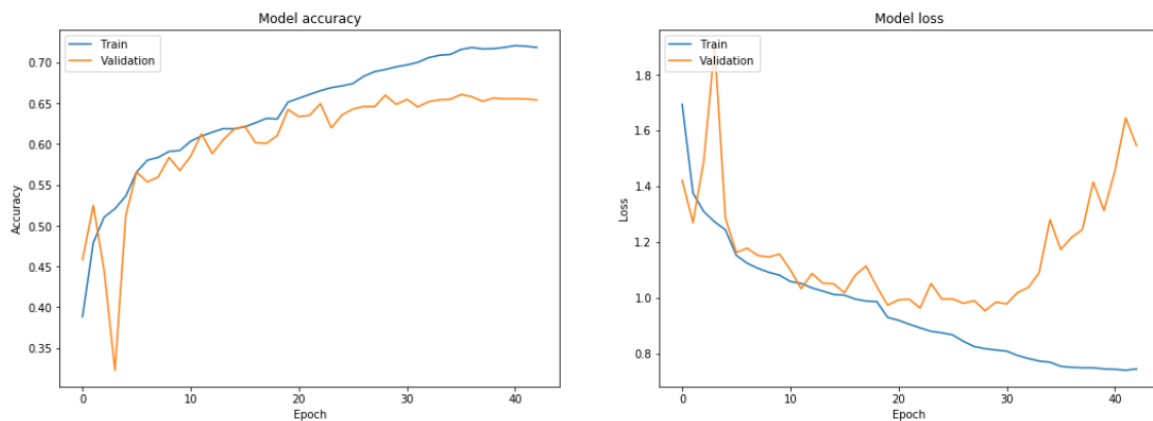
ResNet101

```
model = Sequential()

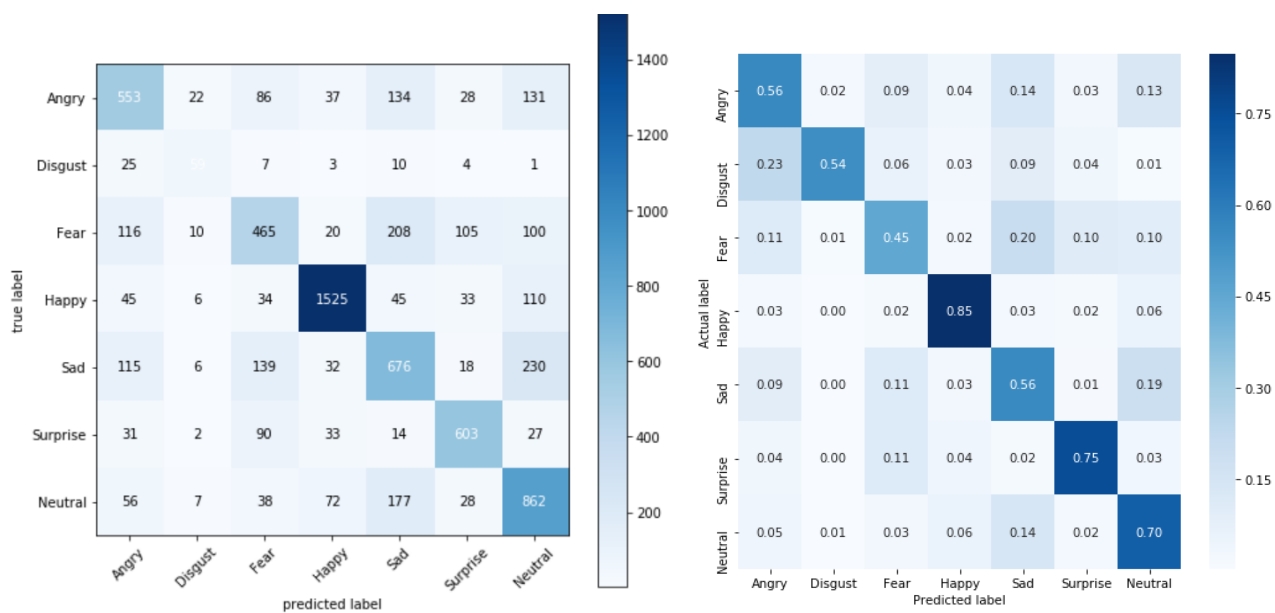
pretrained_model = ResNet101(include_top=False,
                              input_shape=(IMG_WIDTH, IMG_HEIGHT, 3), classes=7,
                              weights='../input/resnet101/resnet101_weights_tf_
                              dim_ordering_tf_kernels_notop.h5')
#for layer in pretrained_model.layers:
#    layer.trainable=False

model.add(pretrained_model)
#model.add(Flatten())
model.add(GlobalAveragePooling2D())
model.add(Dropout(0.2))
# Output layer
model.add(Dense(7, activation='softmax'))
# Compile the model
model.compile(optimizer=Adam(learning_rate=0.0005), loss=tf.keras.losses.SparseCategorical
Crossentropy(), metrics=['accuracy'])
model.summary()
```

Hình 8: Model ResNet101



Hình 9: Accurary và Loss của model ResNet101



Hình 10: Cofusion matrix của model ResNet50

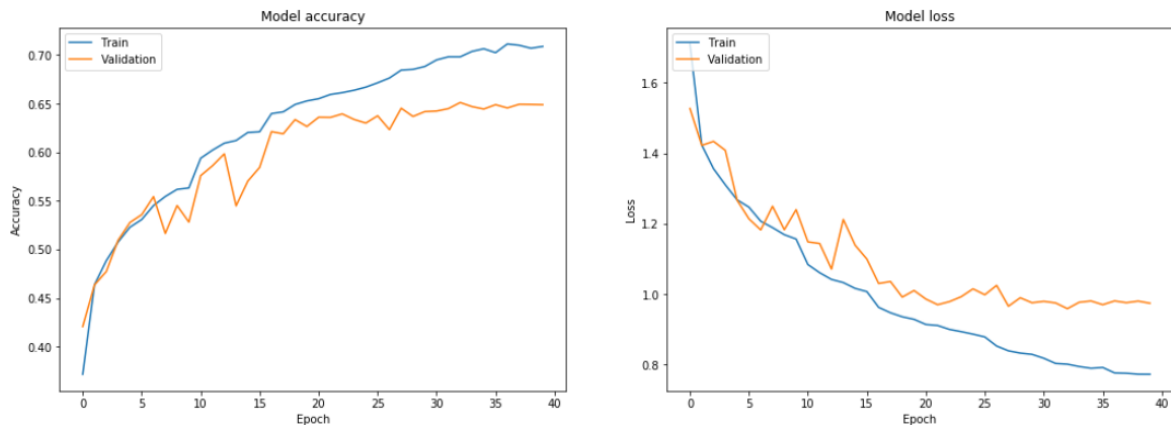
ResNet152

```
model = Sequential()

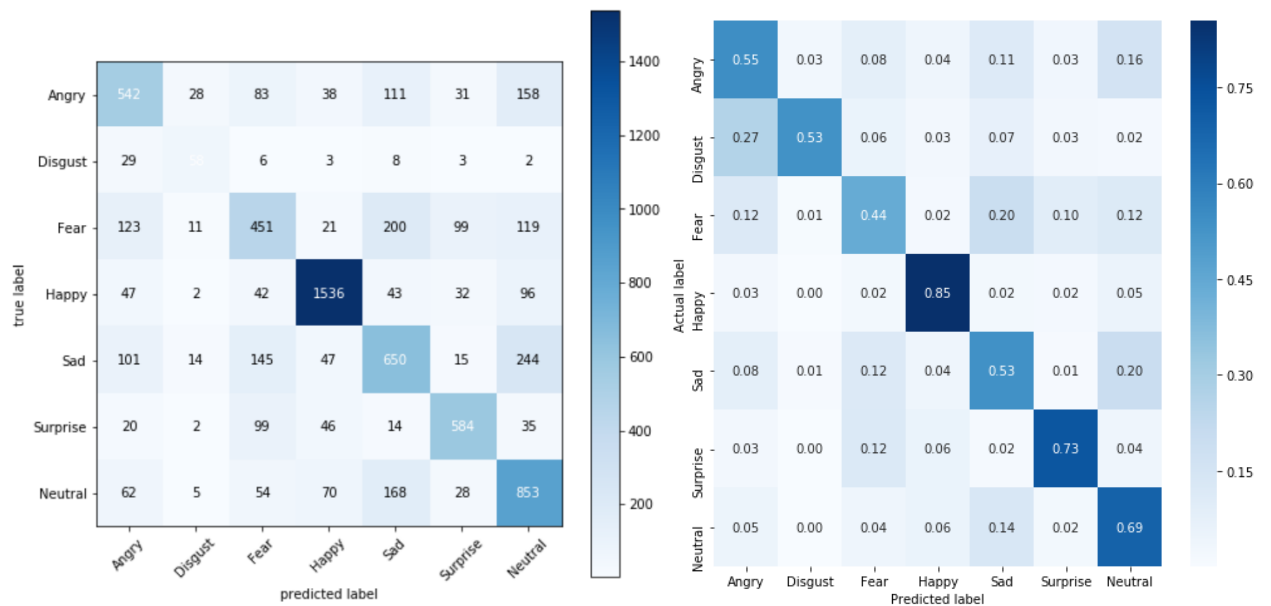
pretrained_model = ResNet152(include_top=False,
                              input_shape=(IMG_WIDTH, IMG_HEIGHT, 3), classes=7,
                              weights='../input/resnet152/resnet152_weights_tf_di
m_ordering_tf_kernels_notop.h5')
#for layer in pretrained_model.layers:
#    layer.trainable=False

model.add(pretrained_model)
#model.add(Flatten())
model.add(GlobalAveragePooling2D())
model.add(Dropout(0.2))
# Output layer
model.add(Dense(7, activation='softmax'))
# Compile the model
model.compile(optimizer=Adam(learning_rate=0.0005), loss=tf.keras.losses.SparseCategoricalCr
ossentropy(), metrics=['accuracy'])
model.summary()
```

Hình 11: Model của ResNet152



Hình 12: Accuracy và Loss của ResNet152



Hình 13: Cofusion matrix của model ResNet50

2.4.3. So sánh 3 model:

	ResNet-50	ResNet-101	ResNet-152
Accuracy	0.75	0.7208	0.7102
Loss	0.8445	0.7324	0.7710
Val_accuracy	0.6314	0.6315	0.6509
Val_loss	0.9885	1.0180	0.9899
Thời gian	Khoảng 40 phút	Khoảng 1 giờ	Khoảng 1 giờ 15 phút

ResNet-152 có *val_accuracy* cao nhất trong 3 model (0.6509), nhưng ResNet-50 có sự ổn định hơn khi có *val_loss* thấp nhất trong 3 model (0.9885) và *val_accuracy* là 0.6314.

2.5. Hướng phát triển

- Tăng số lượng mẫu của các biểu cảm để hệ thống đánh giá tốt hơn và chính xác hơn

- Tăng độ chính xác của mô hình bằng thuật toán Deep Learning cũng như triển khai đến những mô hình mạng học sâu tiên tiến hơn như EfficientNetV1, EfficientNetV2, ...
- Cải thiện web API.
- Hướng đến phát hiện biểu cảm khuôn mặt thông qua video.
- Thu thập thêm tập dữ liệu người Châu Á để model nhận diện biểu cảm khuôn mặt của người Châu Á tốt hơn vì tập dữ liệu *FER-2013* là người Châu Âu.

2.6. Demo:

2.6.1. Xây dựng web API:

Sử dụng thư viện Streamlit để xây dựng web API. Trang web được sử dụng để người dùng có thể upload một ảnh lên, sau đó trang web sẽ dự đoán được biểu cảm gương mặt trong bức ảnh và in ra màn hình kèm theo độ chính xác.

2.6.2. Hình ảnh Demo:



3. KẾT LUẬN

- Cả 3 model đều nhận diện được biểu cảm của người với với độ chính xác trên 60%.
- Kết quả thử nghiệm thực tế cho thấy mô hình khá nhạy khi nhận biết cảm xúc Happy, khá kém với cảm xúc Disgust.
- **Fear** với **Angry**, **Neutral** với **Sad** có biểu cảm khá giống nhau nên model thường nhầm lẫn dẫn đến cho ra độ chính xác không cao.
- Việc hầu hết các mô hình được công bố với tập dữ liệu **FER-2013** đều chỉ đạt độ chính xác thấp (dưới 70%), điều này có thể cho thấy bộ dữ liệu này có những yếu tố mất cân bằng hoặc nhiều khi gán nhãn dữ liệu.

4.TÀI LIỆU THAM KHẢO

- [1] [Nghiên cứu nhận dạng biểu cảm khuôn mặt bằng phương pháp học sâu sử dụng kiến trúc ResNet - Tài liệu, ebook, giáo trình, hướng dẫn \(timtailieu.vn\)](#)
- [2] [FER2013 | Kaggle](#)
- [3] [facial-emotion-recognition/emotion_recognizer.ipynb at main · esra-polat/facial-emotion-recognition · GitHub](#)
- [4] [37-151.pdf \(vap.ac.vn\)](#)
- [5] [1804.08348.pdf \(arxiv.org\)](#)
- [6] [Streamlit documentation](#)

PHỤ LỤC PHÂN CÔNG NHIỆM VỤ

STT	Thành viên	Nhiệm vụ
1	Nguyễn Thị Như Ý	Train model ResNet-101, làm Powerpoint
2	Nguyễn Tấn Tú	Train model ResNet-50, Detect face and Predict
3	Đinh Hoàng Linh Đan	Deploy model lên web API
4	Trần Nguyễn Quỳnh Anh	Train model ResNet-152, làm Powerpoint