Face Emotion Detection

Abeer Almdani | Rana Alzahrani

Table of contents

Introduction

This project will be about building the neural network which classifies the human face images into 5 categories(happiness, neutral, sadness, anger, surprise, disgust, fear) using deep learning models and classification techniques

”Emotions are psychological states brought on by neurophysiological changes, variously associated with thoughts, feelings, behavioural responses, and a degree of pleasure or displeasure.”

— Panksepp, Jaak

Data Sample

Data size 35,685 examples of 48x48 pixel gray scale images of faces divided into train, validation and test dataset.

Data source: Kaggle [1]

Angry

Disgusted

Fearful

Happy

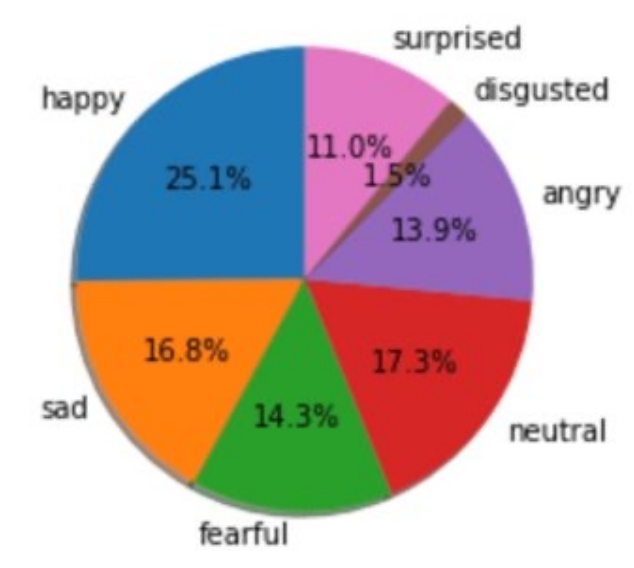
Neutral

Sad

Surprised

Data Analysis

Split Data
Train: 80%
Validation: 15%
Test: 5%



Preprocessing

Rescaling & Resizing

Rescaling (1./255)

Resizing (224,224)

Flip & Rotation

Random Flip(horizontal)

Random Rotation (0.2)

Zoom range

(0.2)

Baseline

NN Baseline 1 = Dense(8) , relu, adam
NN Baseline 2 = Dense(4,8,16,32) , relu, adam
NN Baseline 3 = with scaling, Dense(128, 64, 32) , tanh, adam
NN Baseline 4 = without scaling, Dense(128, 64, 32) , tanh, adam
CNN Baseline = Conv2D(10,5), MaxPooling2D , relu , adam

Model	Training		Validation		Train/ Validation Diff
	Accuracy	Loss	Accuracy	Loss	
NN Baseline 1	28.1987	1.7572	28.2930	1.7645	-0.0942
NN Baseline 2	26.64402	1.752709	26.17017	1.7598	0.473845
NN Baseline 3	28.0463	1.7570	27.9276	1.76537	0.1187
NN Baseline 4	25.62059	1.80954	25.2827	1.81019	0.3378
CNN Baseline	33.17655	1.67375	31.8774998	1.687617	1.29905343

Models

CNN-v1: epochs=5, Conv2(256,128,64,32,16), softmax
CNN-v2 : epochs=50 ,Conv2(32, 32, 64, 64, 128, 128, 256, 256) , Dense (512, 256, 128, 64, 32) , BatchNormalization, Softmax
CNN-v3 : epochs=100, Conv2(32, 32, 64, 64, 128, 128, 256, 256) , Dense (64,64), BatchNormalization, Early stopping, softmax
CNN-v4 : CNN 3 without Dropout.
CNN-v5 : CNN 3 with Dropout in layer 1,2

	Dropout	Training		Validation		Train/ Validation Diff
		Accuracy	Loss	Accuracy	Loss	
CNN-v1	0.4, 0.3	26.1039	1.7867	26.22237	1.7886	0.1183
CNN-v2	0.1, 0.3	57.2511	1.12516	53.40177	1.1936	3.8493
CNN-v3	0.1, 0.3	75.847	0.6896	59.9443	1.0908	15.9027
CNN-v4	-	95.79	0.1285	55.09	2.2728	40.70
CNN-v5	0.3	61.6671	1.0303	53.4713	1.2339	8.1957

Models Con.

CNN-v6: CNN3 with dropout layers(3,4,5)
CNN-v7: CNN 4 with dropout in layer 6,9
CNN-v8: CNN4 low layer and dropout
CNN-v9 : CNN4 with kernel regularizer =L2, BatchNormalization, SGD optimizer, without kernel_initializer = he_normal

	Dropout	Training		Validation		Train/ Validation Diff
		Accuracy	Loss	Accuracy	Loss	
CNN-v6	0.3	81.42	0.520	57.65	1.473	23.77
CNN-v7	0.3	80.94	0.5173	53.14	1.7264	27.80
CNN-v8	0.5	91.78	0.2308	52.43	2.4780	39.35
CNN-v9	0.7	46.46	1.4107	48.44	1.3506	0.0601

Transfer Learning Models

DenseNet : 10 epochs, layers =Dense(256,128,64,32), BatchNormalization, GlobalAveragePooling2D, relu,softmax
VGG16-v1 : 15 epochs, Dense(125,25) , BatchNormalization , relu, adam , softmax
VGG16-v2 :100 epochs, Conv2D (32),BatchNormalization,MaxPooling2D, Dense (128,64,32),tanh,softmax
VGG19 : 10 epochs, Dense(200,100,50) , GlobalAveragePooling2D,BatchNormalization, relu ,softmax
MobileNet : Flatten, softmax
MobileNet-v2 : Conv2D(128),Dense(1024,100,512),relu ,softmax
ResNet50-v1 : Dense(200,100,100) , GlobalAveragePooling2D,BatchNormalization, relu ,softmax
ResNet50-v2 : Conv2D(64,128), Dense(200,100,50), relu, software

	Dropout	Training		Validation		Train/ Validation Diff
		Accuracy	Loss	Accuracy	Loss	
DenseNet	-	43.39	1.4876	37.01	1.6416	6.38
VGG16-v1		18.66	1.2184	17.91	1.7029	0.0075
VGG16-v2	0.25	30.00	1.7997	28.52	1.7931	1.7931
VGG19	-	32.1836	1.7213	30.4854	1.7467	1.6981
MobileNet -v1	-	43.75	4.9195	47.266	-	3.516
MobileNet-v2	-	26.88	1.7853	34.77	-	7,89
ResNet50-v1	-	28.0245	1.8555	27.6666	27.6666	0.3579
ResNet50-v2	-	39.26	1.5581	15.63	2.2828	0.7247

What is the best category that the image belong to :

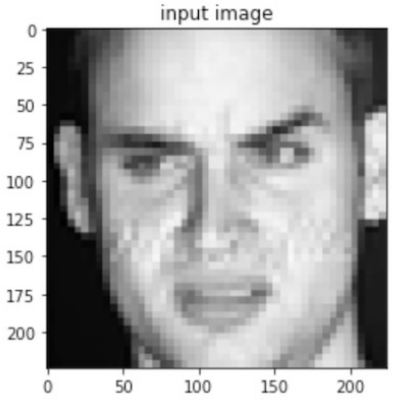
Model Predict : angry
Real Category : disgusted

Model Predict : sad
Real Category : fearful

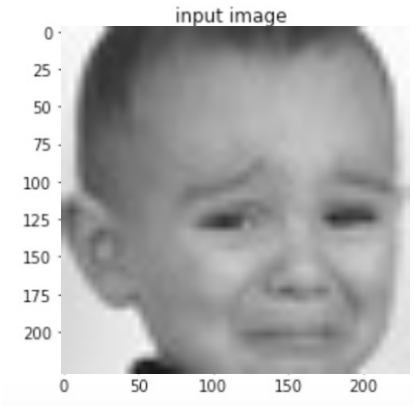
Model Predict : angry
Real Category : neutral

Test Model

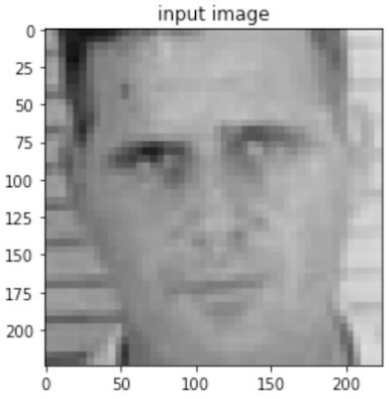
the image is of disgusted



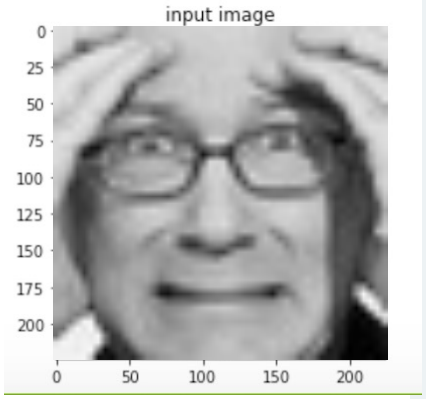
the image is of sad



the image is of neutral

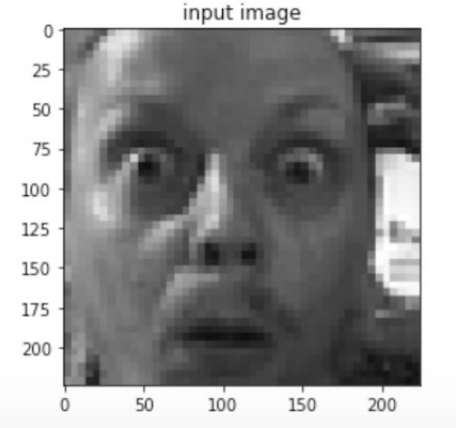


the image is of fearful

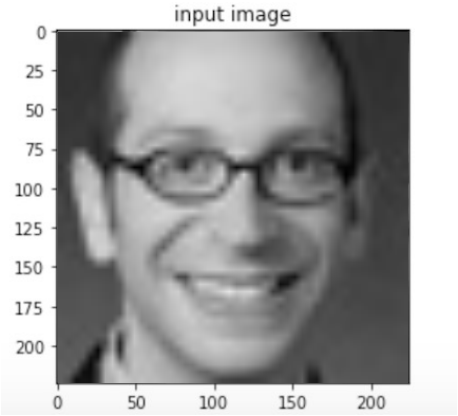


Test Model

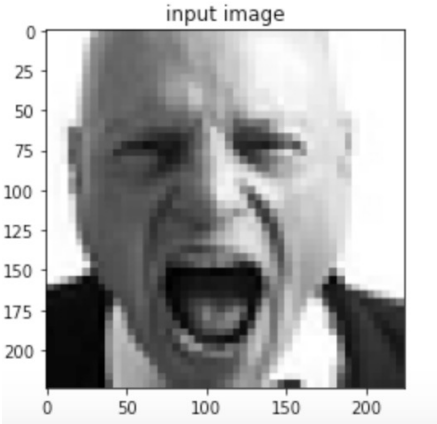
the image is of surprised



the image is of happy



the image is of angry



Challenges

Data Quality Issues

- Watermark
- wrong classification
- Non-descriptive images

Time Issues

Run time about 7-8 h/model

Technical Issues

- Dead kernel
- System Crushed

Remote Working

No direct contact with the team

Comparison with previous projects

EMOTION DETECTOR

Audio Dataset

OUR PROJECT

Dataset is Images

Gray Scale

CNN-v3 is best model

Scores:
Train: 75.84%
Validation: 59.94%

FACIAL EMOTION RECOGNITION

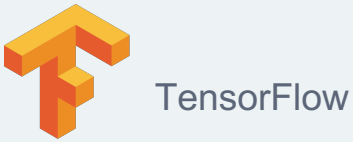
Dataset is Images

Colorful images

VGG16 is best model

Scores:
Train: 47.81%
Validation: 43.08%

Tools



Appendix

Transfer learning
saves training time, gives better performance in most cases, and reduces the need for a huge dataset. It isn't a generalized method but helps in solving related problems

VGG-16 [Visual Geometry Group — 16] is one of the simplest CNN architectures used in ImageNet competitions

The architecture is as follows:
2Conv — 1Maxpool — 2Conv — 1Maxpool — 3Conv — 1Maxpool — 3Conv — 1Maxpool — 3Conv — 1Maxpool — 1FC — 1FC — 1FC

VGG-19 architecture is very much similar to VGG-16. We have 3 additional convolutional layers for the VGG-16 network.

The architecture is as follows:
2Conv — 1Maxpool — 2Conv — 1Maxpool — 4Conv — 1Maxpool — 4Conv — 1Maxpool — 4Conv — 1Maxpool — 1FC — 1FC — 1FC

MobileNet is **a convolutional neural network that is 53 layers deep**. You can load a pretrained version of the network trained on more than a million images from the ImageNet database, The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals [5]

Future Work

Improve model by adding layers to more accuracy.

Merge new data to better learning.

Try more models.

Build an application

Resources

- [1] https://www.kaggle.com/ananthu017/emotion-detection-fer
- [2] https://github.com/shami-am/emotion-recognition/blob/main/Emotion%20Recognition%20-%20Presentation.pptx
- [3] https://www.youtube.com/watch?v=G1Uhs6NVi-M
- [4] https://koushik1102.medium.com/transfer-learning-with-vgg16-and-vgg19-the-simpler-way-ad4eec1e2997
- [5]https://www.google.com/search?q=what+is+moble+net+model&oq=what+is+moble+net+model&aqs=chrome..69i57j0i10i19.17004j0j15&sourceid=chrome&ie=UTF-8

Thank You!
Any Question?