Face Emotion Detection using Deep Learning

Abeer Almdani | Rana Alzahrani

Introduction

This project will be about building the neural network which classifies the human face images into 7 categories (happy, neutral, sad, anger, surprise, disgust, fear) using deep learning models and classification techniques.

Data

The dataset contains 35,685 examples of 48x48 pixel gray scale images of faces divided into train and test dataset. Images are categorized based on the emotion shown in the facial expressions (happiness, neutral, sadness, anger, surprise, disgust, fear).

Data source [https://www.kaggle.com/ananthu017/emotion-detection-fer]

Tools

- Software Platform
    i. Jupyter Notebook
- Programming Language
    i. Python
- Python Libraries:
    i. Tensorflow
    ii. Keras
    iii. Pandas
    iv. Numpy
    v. Matplotlib
    vi. Seaborn

Algorithms and Models

2. Exploratory Data Analysis

- Read the data
- Rescaling & Resizing
- Flip & Rotation
- Zoom range

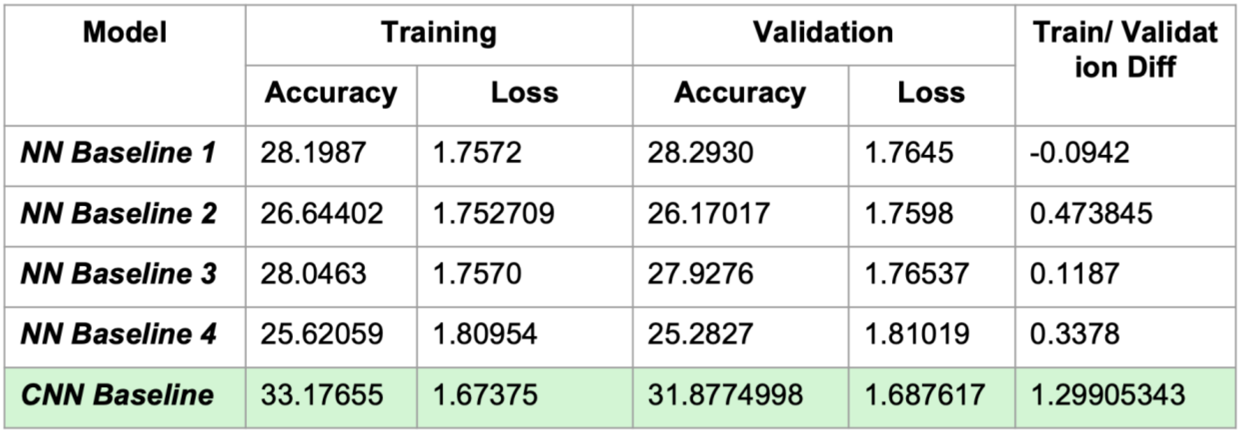
3. Deep Learning

- **Base Line**
  - **NN:** [train=28%, valid = 27%]
  - **CNN:** [train=33%, valid = 31%]

- **Models**
  - NN
  - CNN
    - We do 10 CNN to improve we model
    - **CNN-v3**: [train=75%, valid = 60%] **the best**
- **Transfer learning**
  - DenseNe
  - VGG16
  - VGG19
  - MobileNet
    - MobileNet : [train=43%, valid=47%] **the best**
  - ResNet50

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

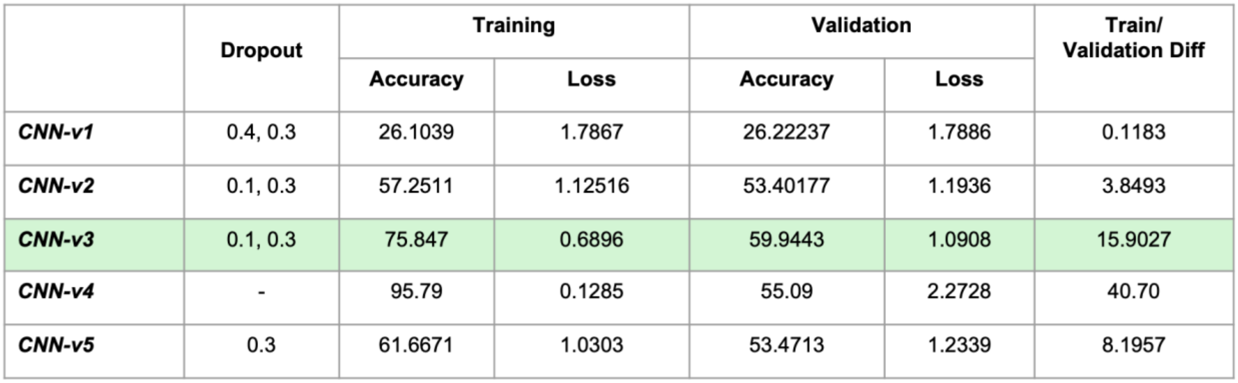


Model	Training		Validation		Train/ Validation Diff
	Accuracy	Loss	Accuracy	Loss	
NN Baseline 1	28.1987	1.7572	28.2930	1.7645	-0.0942
NN Baseline 2	26.64402	1.752709	26.17017	1.7598	0.473845
NN Baseline 3	28.0463	1.7570	27.9276	1.76537	0.1187
NN Baseline 4	25.62059	1.80954	25.2827	1.81019	0.3378
CNN Baseline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

Model Architectures and Scores

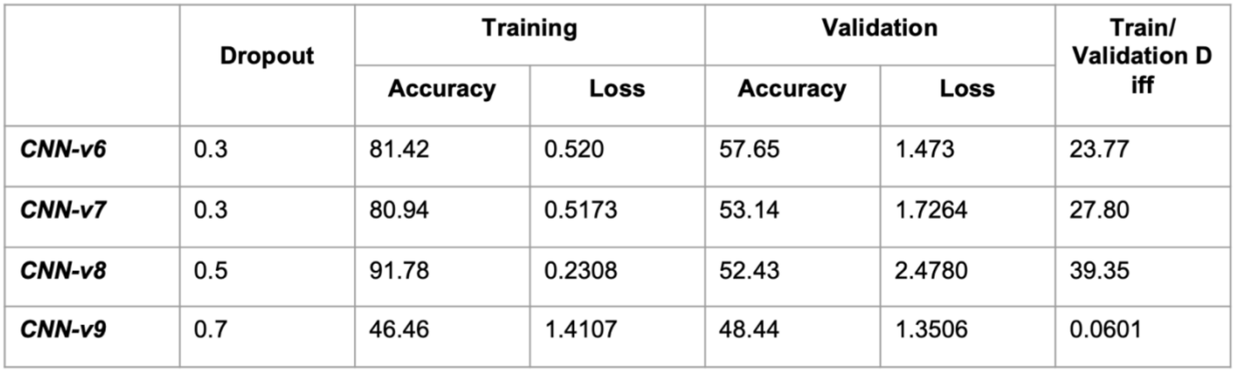
- CNN-v1: epochs=5, Conv2(256,128,64,32,16), softmax
- CNN-v2 : epochs=50 ,Conv2(32, 32, 64, 64, 128, 128, 256, 256) , Dense (512, 256, 128, 64, 32) , BatchNormalization, Softmax
- CNN-v3 : epochs=100, Conv2(32, 32, 64, 64, 128, 128, 256, 256) , Dense (64,64), BatchNormalization, Early stopping, softmax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



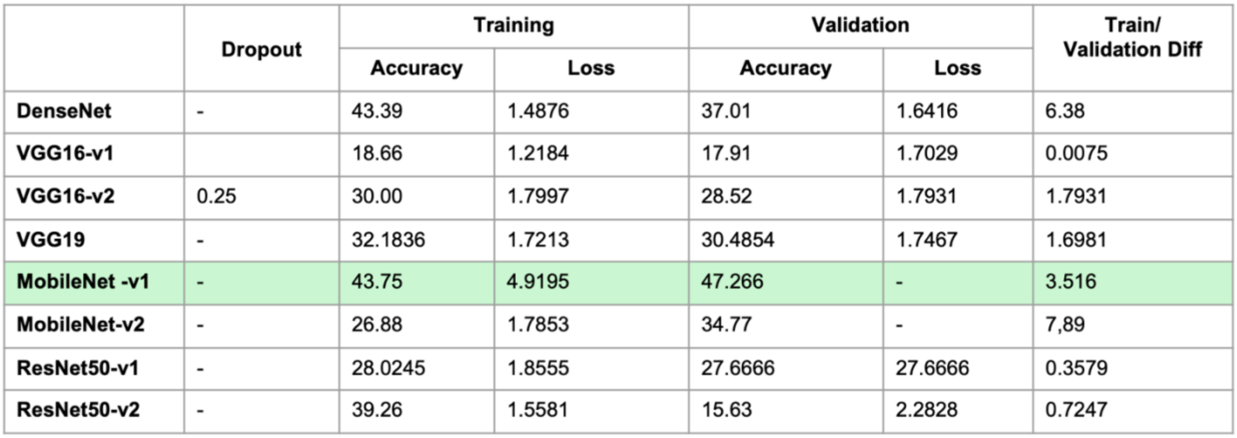
	Dropout	Training		Validation		Train/ Validation Diff
		Accuracy	Loss	Accuracy	Loss	
CNN-v1	0.4, 0.3	26.1039	1.7867	26.22237	1.7886	0.1183
CNN-v2	0.1, 0.3	57.2511	1.12516	53.40177	1.1936	3.8493
CNN-v3	0.1, 0.3	75.847	0.6896	59.9443	1.0908	15.9027
CNN-v4	-	95.79	0.1285	55.09	2.2728	40.70
CNN-v5	0.3	61.6671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

Figure1: Best Models' Scores



	Dropout	Training		Validation		Train/ Validation D iff
		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

Figure2: Models' Scores.



	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

- Figure3: Best Transfer Learning Models' Scores