

ASSIGNMENT: 2

Assignment Document

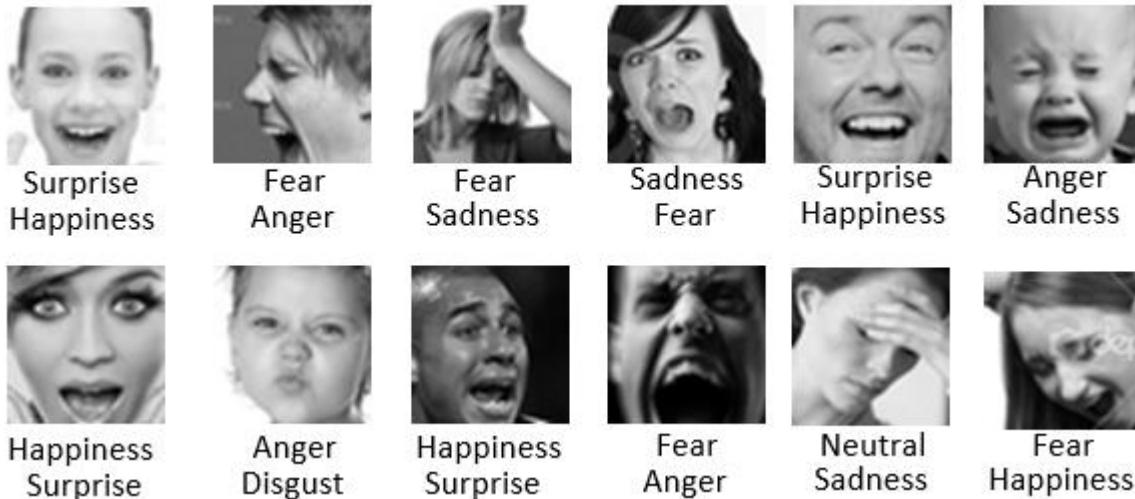
edureka!

edureka!

© Brain4ce Education Solutions Pvt. Ltd.

Scenario - 1

Problem Statement:



The data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centred and occupies about the same amount of space in each image. The task is to categorize each face based on the emotion shown in the facial expression into one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral).

[Fer2013.csv](#) contains two columns, "emotion" and "pixels". The "emotion" column contains a numeric code ranging from 0 to 6, inclusive, for the emotion that is present in the image. The "pixels" column contains a string surrounded in quotes for each image. The contents of this string a space-separated pixel values in row major order.

The dataset consists of 28,709 examples.

This dataset was prepared by Pierre-Luc Carrier and Aaron Courville, as part of an ongoing research project. They have

graciously provided the workshop organizers with a preliminary version of their dataset to use for this contest.

Note: Please use google colab to work on this project. Also, make sure to select GPU backend while selecting a runtime.

<https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d>

** Give some aspect of the notebook for them to have part of code available.

Question 1:

Reshape and Normalize the data:

HINT:

- Split pixels by space to get columns
- reshape the input image
 - 48, 48 ,1
- change the type of data to float32

Question 2:

Normalize & Train_Test Split

HINT:

- Normalize the data by dividing with 255
- Split data into train and test (90,10)

Question 3:

Define CNN Model:

- Layer 1
 - 2 Conv2d with 64 filters of 5,5 filter
 - BatchNormalization layer
 - Max Pooling layer with 2,2
 - Activation relu
- Layer 2
 - 2 Conv2d with 128 filters of 5,5 filter
 - BatchNormalization layer
 - Max Pooling layer with 2,2
 - Activation relu
- Layer 3
 - 2 Conv2d with 256 filters of 3,3 filter
 - BatchNormalization layer
 - Max Pooling layer with 2,2
 - Activation relu
- Layer 4
 - Flatten layer
 - Dense Layer with 128 Neuron
 - BatchNormalization
 - Activation Relu
 - Dropout 0.25
 - Dense 7 neurons with Softmax
- Loss: Categorical cross-entropy
- Optimizer: Adam

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 48, 48, 64)	1664
conv2d_2 (Conv2D)	(None, 48, 48, 64)	102464
batch_normalization_1 (Batch	(None, 48, 48, 64)	256

max_pooling2d_1 (MaxPooling2D)	(None, 24, 24, 64)	0
conv2d_3 (Conv2D)	(None, 24, 24, 128)	204928
conv2d_4 (Conv2D)	(None, 24, 24, 128)	409728
batch_normalization_2 (BatchNormalization)	(None, 24, 24, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 12, 12, 128)	0
conv2d_5 (Conv2D)	(None, 12, 12, 256)	295168
conv2d_6 (Conv2D)	(None, 12, 12, 256)	590080
batch_normalization_3 (BatchNormalization)	(None, 12, 12, 256)	1024
max_pooling2d_3 (MaxPooling2D)	(None, 6, 6, 256)	0
flatten_1 (Flatten)	(None, 9216)	0
dense_1 (Dense)	(None, 128)	1179776
batch_normalization_4 (BatchNormalization)	(None, 128)	512
activation_1 (Activation)	(None, 128)	0
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 7)	903
activation_2 (Activation)	(None, 7)	0

=====

Total params: 2,787,015
Trainable params: 2,785,863
Non-trainable params: 1,152

Question 4:

Fit CNN Model:

HINT:

- Epoch = 20
- Batch_size 64
- Lr = 1e-3

Question 5:

Draw confusion matrix for the trained model.

HINT:

- Use `model.predict_classes`
- Use `np.argmax(y_test, axis=1)`

Question 6:

Save the model to external .h5 File

Question 7:

Convert the saved model to .pb model

HINT:

- Use the `free_session` function to convert
- Create keras session: `sess = K.get_session()`
- Use `tf.train.write_graph` to convert frozen graph as .pb

