In [1]:
```
import numpy as np
import matplotlib.pyplot as plt

image_height = 48
image_width = 48
emotions_count = 8
emotion_labels = ['neutral', 'happiness', 'surprise', 'sadness', 'anger', 'disgust', 'fear', 'contempt']

samples = 35393 # 2~35394
training_samples = 28317  # 2~28318 (Training)
validation_samples = 3541 # 28319~31859 (PublicTest)
test_samples = 3535       # 31860~35394 (PrivateTest)
```

In [2]:
```
import tensorflow as tf
from tensorflow.python.keras.layers import Dense, GlobalAveragePooling2D, MaxPool2D, Input, Conv2D, Flatten
from tensorflow.python.keras.models import Model
from tensorflow.python.keras import layers, Sequential,losses, metrics
from tensorflow.python.keras import optimizers, callbacks, models
from tensorflow.python.keras.optimizer_v2 import adam
```

```
2021-12-28 01:39:27.607036: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library libcudart.so.11.0
```

In [3]:
```
image_path = "./dataset/images.npy"
emotion_multi_path = "./dataset/emotions_multi.npy"
emotion_single_path = "./dataset/emotions_single.npy"

images = np.load(image_path)
emotions_multi = np.load(emotion_multi_path)
emotions_single = np.load(emotion_single_path)

print(images.shape)
print(emotions_multi.shape)
print(emotions_single.shape)
```

```
(35393, 48, 48, 1)
(35393, 8)
(35393, 8)
```

In [4]:
```
#emotions = emotions_single
emotions = emotions_multi

images = tf.convert_to_tensor(images)
#images = tf.image.grayscale_to_rgb(images)
emotions = tf.convert_to_tensor(emotions)
print("images shape:", images.shape)
print("emotions shape:", emotions.shape)
```

```
images shape: (35393, 48, 48, 1)
emotions shape: (35393, 8)
2021-12-28 01:39:32.069569: I tensorflow/compiler/jit/xla_cpu_device.cc:41] Not creating XLA devices, tf_xla_enable_xla_devices no
t set
2021-12-28 01:39:32.071359: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcuda.so.1
2021-12-28 01:39:32.139181: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:941] successful NUMA node read from SysFS had n
egative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-12-28 01:39:32.139850: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1720] Found device 0 with properties:
pciBusID: 0000:05:00.0 name: GeForce RTX 2080 Ti computeCapability: 7.5
coreClock: 1.545GHz coreCount: 68 deviceMemorySize: 10.76GiB deviceMemoryBandwidth: 573.69GiB/s
2021-12-28 01:39:32.139894: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcudart.so.11.0
2021-12-28 01:39:32.146371: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcublas.so.11
2021-12-28 01:39:32.146454: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcublasLt.so.11
2021-12-28 01:39:32.149717: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcufft.so.10
2021-12-28 01:39:32.151153: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcurand.so.10
2021-12-28 01:39:32.160915: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcusolver.so.10
2021-12-28 01:39:32.163359: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcusparse.so.11
2021-12-28 01:39:32.164496: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcudnn.so.8
2021-12-28 01:39:32.164740: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:941] successful NUMA node read from SysFS had n
egative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-12-28 01:39:32.165850: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:941] successful NUMA node read from SysFS had n
egative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2021-12-28 01:39:32.166542: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1862] Adding visible gpu devices: 0
2021-12-28 01:39:32.168468: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI D
eep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations:  AVX2 AVX512F FMA
```

In [5]:
```
from tensorflow.python.keras import layers
# choose one method:
images = layers.Rescaling(1./127.5, offset= -1)(images)
```

In [6]:
```
training_size = training_samples + validation_samples
test_size = test_samples

training_images = images[:training_size]
test_images = images[training_size:]
training_emotions = emotions[:training_size]
test_emotions = emotions[training_size:]

print("training_images shape:", training_images.shape)
print("training_emotions shape:", training_emotions.shape)
print("test_images shape:", test_images.shape)
print("test_emotions shape:", test_emotions.shape)
```

```
training_images shape: (31858, 48, 48, 1)
training_emotions shape: (31858, 8)
test_images shape: (3535, 48, 48, 1)
test_emotions shape: (3535, 8)
```

In [7]:
```
from tensorflow.python.keras import losses, metrics
from tensorflow.python.keras.optimizer_v2 import adam

cce = losses.CategoricalCrossentropy()
mse = losses.MeanSquaredError()

tf.config.run_functions_eagerly(True)
def model_acc(y_true, y_pred):
    size = y_true.shape[0]
    acc = 0
    for i in range(size):
        true = y_true[i]
        pred = y_pred[i]
        index_max = tf.argmax(pred).numpy()
        if true[index_max].numpy()==tf.reduce_max(true).numpy():
            acc += 1
    return acc/size
```

In [8]:

```
from tensorflow.python.keras.applications import vgg16, resnet_v2, densenet, efficientnet
from tensorflow.python.keras.layers import Dense, GlobalAveragePooling2D, MaxPool2D, Input, Conv2D, Flatten, Concatenate, Dropout
from tensorflow.python.keras.models import Model
from tensorflow.python.keras import layers, Sequential
from tensorflow.keras import backend as K

base_model = vgg16.VGG16(include_top=False,
                         weights="imagenet",
                         input_shape=(48,48,3))
base_model.trainable=True

# base_model.set_weights("vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5")
'''
model = keras.Sequential()
model.add(base_model.layers[2])
model.add()
feat1 = GlobalAveragePooling2D()(input_layer)
model.add(base_model.layers[2])
'''

''' Runnable
input_layer = Input(shape=(48,48,3))
print(input_layer.shape)
feat1 = GlobalAveragePooling2D()(input_layer)
print("feature1", feat1.shape)
x = base_model.layers[0](input_layer)
x = base_model.layers[1](x)
'''

input_layer = Input(shape=(48,48,3))
print(input_layer.shape)
feat1 = GlobalAveragePooling2D()(input_layer)
print("feature1", feat1.shape)
x = base_model.layers[1](input_layer)

x = base_model.layers[2](x)
x = base_model.layers[3](x)
print(x.shape)
feat2 = GlobalAveragePooling2D()(x)
print("feature2", feat2.shape)

x = base_model.layers[4](x)
x = base_model.layers[5](x)
x = base_model.layers[6](x)
```

```
print(x.shape)
feat3 = GlobalAveragePooling2D()(x)
print("feature3", feat3.shape)

x = base_model.layers[7](x)
x = base_model.layers[8](x)
x = base_model.layers[9](x)
x = base_model.layers[10](x)
print(x.shape)
feat4 = GlobalAveragePooling2D()(x)
print("feature4", feat4.shape)

x = base_model.layers[11](x)
x = base_model.layers[12](x)
x = base_model.layers[13](x)
x = base_model.layers[14](x)
print(x.shape)
feat5 = GlobalAveragePooling2D()(x)
print("feature5", feat5.shape)

x = base_model.layers[15](x)
x = base_model.layers[16](x)
x = base_model.layers[17](x)
x = base_model.layers[18](x)
print(x.shape)
feat6 = GlobalAveragePooling2D()(x)
print("feature6", feat6.shape)

x = tf.concat([feat1, feat2, feat3, feat4, feat5, feat6], -1)
print("combined feature", x.shape)

x = Dense(units=4096, activation='relu')(x)
x = Dropout(0.5)(x)
x = Dense(units=4096, activation='relu')(x)
x = Dropout(0.5)(x)
output_layer = Dense(units=8, activation='softmax')(x)

model = Model(inputs=input_layer, outputs=output_layer)

model.summary()


model.compile(optimizer=adam.Adam(learning_rate=3e-5),
              loss=mse,
```

```
              metrics = [model_acc])

model.fit(x=tf.image.grayscale_to_rgb(training_images),
          y=training_emotions,
          batch_size=32,
          epochs=20,
          validation_data=(tf.image.grayscale_to_rgb(test_images), test_emotions))

model.compile(optimizer=adam.Adam(learning_rate=1e-5),
              loss=mse,
              metrics = [model_acc])

model.fit(x=tf.image.grayscale_to_rgb(training_images),
          y=training_emotions,
          batch_size=32,
          epochs=20,
          validation_data=(tf.image.grayscale_to_rgb(test_images), test_emotions))
```

```
(None, 48, 48, 3)
feature1 (None, 3)
(None, 24, 24, 64)
feature2 (None, 64)
(None, 12, 12, 128)
feature3 (None, 128)
(None, 6, 6, 256)
feature4 (None, 256)
(None, 3, 3, 512)
feature5 (None, 512)
(None, 1, 1, 512)
feature6 (None, 512)
combined feature (None, 1475)
Model: "model"
```

| Layer (type) | Output Shape | Param # | Connected to |
|---|---|---|---|
| input_2 (InputLayer) | [(None, 48, 48, 3)] | 0 | |
| block1_conv1 (Conv2D) | (None, 48, 48, 64) | 1792 | input_2[0][0] |
| block1_conv2 (Conv2D) | (None, 48, 48, 64) | 36928 | block1_conv1[1][0] |
| block1_pool (MaxPooling2D) | (None, 24, 24, 64) | 0 | block1_conv2[1][0] |

| | | | |
|---|---|---|---|
| block2_conv1 (Conv2D) | (None, 24, 24, 128) | 73856 | block1_pool[1][0] |
| block2_conv2 (Conv2D) | (None, 24, 24, 128) | 147584 | block2_conv1[1][0] |
| block2_pool (MaxPooling2D) | (None, 12, 12, 128) | 0 | block2_conv2[1][0] |
| block3_conv1 (Conv2D) | (None, 12, 12, 256) | 295168 | block2_pool[1][0] |
| block3_conv2 (Conv2D) | (None, 12, 12, 256) | 590080 | block3_conv1[1][0] |
| block3_conv3 (Conv2D) | (None, 12, 12, 256) | 590080 | block3_conv2[1][0] |
| block3_pool (MaxPooling2D) | (None, 6, 6, 256) | 0 | block3_conv3[1][0] |
| block4_conv1 (Conv2D) | (None, 6, 6, 512) | 1180160 | block3_pool[1][0] |
| block4_conv2 (Conv2D) | (None, 6, 6, 512) | 2359808 | block4_conv1[1][0] |
| block4_conv3 (Conv2D) | (None, 6, 6, 512) | 2359808 | block4_conv2[1][0] |
| block4_pool (MaxPooling2D) | (None, 3, 3, 512) | 0 | block4_conv3[1][0] |
| block5_conv1 (Conv2D) | (None, 3, 3, 512) | 2359808 | block4_pool[1][0] |
| block5_conv2 (Conv2D) | (None, 3, 3, 512) | 2359808 | block5_conv1[1][0] |
| block5_conv3 (Conv2D) | (None, 3, 3, 512) | 2359808 | block5_conv2[1][0] |
| block5_pool (MaxPooling2D) | (None, 1, 1, 512) | 0 | block5_conv3[1][0] |
| global_average_pooling2d (Globa | (None, 3) | 0 | input_2[0][0] |
| global_average_pooling2d_1 (Glo | (None, 64) | 0 | block1_pool[1][0] |
| global_average_pooling2d_2 (Glo | (None, 128) | 0 | block2_pool[1][0] |
| global_average_pooling2d_3 (Glo | (None, 256) | 0 | block3_pool[1][0] |
| global_average_pooling2d_4 (Glo | (None, 512) | 0 | block4_pool[1][0] |
| global_average_pooling2d_5 (Glo | (None, 512) | 0 | block5_pool[1][0] |
| tf.concat (TFOpLambda) | (None, 1475) | 0 | global_average_pooling2d[0][0]<br>global_average_pooling2d_1[0][0] |

```
                                                    global_average_pooling2d_2[0][0]
                                                    global_average_pooling2d_3[0][0]
                                                    global_average_pooling2d_4[0][0]
                                                    global_average_pooling2d_5[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 4096)         6045696     tf.concat[0][0]
__________________________________________________________________________________________________
dropout (Dropout)               (None, 4096)         0           dense[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 4096)         16781312    dropout[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 4096)         0           dense_1[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)                 (None, 8)            32776       dropout_1[0][0]
==================================================================================================
Total params: 37,574,472
Trainable params: 37,574,472
Non-trainable params: 0
__________________________________________________________________________________________________
/userhome/cs/fym666/anaconda3/envs/tensorflow/lib/python3.8/site-packages/tensorflow/python/data/ops/dataset_ops.py:3503: UserWarn
ing: Even though the tf.config.experimental_run_functions_eagerly option is set, this option does not apply to tf.data functions.
tf.data functions are still traced and executed as graphs.
  warnings.warn(
2021-12-28 01:39:35.470615: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:116] None of the MLIR optimization passes a
re enabled (registered 2)
2021-12-28 01:39:35.471169: I tensorflow/core/platform/profile_utils/cpu_utils.cc:112] CPU Frequency: 2199995000 Hz
2021-12-28 01:39:35.496348: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcudnn.so.8
Epoch 1/20
2021-12-28 01:39:37.864090: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcublas.so.11
2021-12-28 01:39:38.447973: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcublasLt.so.11
996/996 [==============================] - 95s 92ms/step - loss: 0.1094 - model_acc: 0.3867 - val_loss: 0.0297 - val_model_acc: 0.
6709
Epoch 2/20
996/996 [==============================] - 88s 89ms/step - loss: 0.0268 - model_acc: 0.7106 - val_loss: 0.0229 - val_model_acc: 0.
7324
Epoch 3/20
996/996 [==============================] - 90s 91ms/step - loss: 0.0202 - model_acc: 0.7673 - val_loss: 0.0185 - val_model_acc: 0.
7763
Epoch 4/20
996/996 [==============================] - 90s 90ms/step - loss: 0.0157 - model_acc: 0.8148 - val_loss: 0.0182 - val_model_acc: 0.
```

```
7844
Epoch 5/20
996/996 [==============================] - 92s 92ms/step - loss: 0.0131 - model_acc: 0.8444 - val_loss: 0.0175 - val_model_acc: 0.
7966
Epoch 6/20
996/996 [==============================] - 91s 92ms/step - loss: 0.0110 - model_acc: 0.8622 - val_loss: 0.0177 - val_model_acc: 0.
7881
Epoch 7/20
996/996 [==============================] - 90s 91ms/step - loss: 0.0092 - model_acc: 0.8867 - val_loss: 0.0170 - val_model_acc: 0.
7994
Epoch 8/20
996/996 [==============================] - 92s 92ms/step - loss: 0.0079 - model_acc: 0.9038 - val_loss: 0.0163 - val_model_acc: 0.
8062
Epoch 9/20
996/996 [==============================] - 94s 94ms/step - loss: 0.0069 - model_acc: 0.9170 - val_loss: 0.0170 - val_model_acc: 0.
7932
Epoch 10/20
996/996 [==============================] - 94s 94ms/step - loss: 0.0063 - model_acc: 0.9216 - val_loss: 0.0149 - val_model_acc: 0.
8230
Epoch 11/20
996/996 [==============================] - 90s 90ms/step - loss: 0.0055 - model_acc: 0.9320 - val_loss: 0.0143 - val_model_acc: 0.
8239
Epoch 12/20
996/996 [==============================] - 90s 91ms/step - loss: 0.0049 - model_acc: 0.9387 - val_loss: 0.0148 - val_model_acc: 0.
8230
Epoch 13/20
996/996 [==============================] - 92s 92ms/step - loss: 0.0045 - model_acc: 0.9436 - val_loss: 0.0139 - val_model_acc: 0.
8248
Epoch 14/20
996/996 [==============================] - 86s 86ms/step - loss: 0.0043 - model_acc: 0.9457 - val_loss: 0.0144 - val_model_acc: 0.
8239
Epoch 15/20
996/996 [==============================] - 82s 83ms/step - loss: 0.0038 - model_acc: 0.9503 - val_loss: 0.0138 - val_model_acc: 0.
8261
Epoch 16/20
996/996 [==============================] - 89s 89ms/step - loss: 0.0036 - model_acc: 0.9557 - val_loss: 0.0137 - val_model_acc: 0.
8284
Epoch 17/20
996/996 [==============================] - 89s 89ms/step - loss: 0.0034 - model_acc: 0.9551 - val_loss: 0.0140 - val_model_acc: 0.
8242
Epoch 18/20
996/996 [==============================] - 92s 92ms/step - loss: 0.0032 - model_acc: 0.9614 - val_loss: 0.0137 - val_model_acc: 0.
8309
Epoch 19/20
```

```
996/996 [==============================] - 87s 88ms/step - loss: 0.0030 - model_acc: 0.9627 - val_loss: 0.0139 - val_model_acc: 0.
8309
Epoch 20/20
996/996 [==============================] - 91s 91ms/step - loss: 0.0028 - model_acc: 0.9630 - val_loss: 0.0138 - val_model_acc: 0.
8326
Epoch 1/20
996/996 [==============================] - 87s 88ms/step - loss: 0.0023 - model_acc: 0.9716 - val_loss: 0.0133 - val_model_acc: 0.
8346
Epoch 2/20
996/996 [==============================] - 91s 91ms/step - loss: 0.0019 - model_acc: 0.9811 - val_loss: 0.0133 - val_model_acc: 0.
8343
Epoch 3/20
996/996 [==============================] - 91s 91ms/step - loss: 0.0017 - model_acc: 0.9849 - val_loss: 0.0136 - val_model_acc: 0.
8340
Epoch 4/20
996/996 [==============================] - 89s 89ms/step - loss: 0.0016 - model_acc: 0.9862 - val_loss: 0.0137 - val_model_acc: 0.
8326
Epoch 5/20
996/996 [==============================] - 90s 90ms/step - loss: 0.0016 - model_acc: 0.9870 - val_loss: 0.0140 - val_model_acc: 0.
8368
Epoch 6/20
996/996 [==============================] - 81s 81ms/step - loss: 0.0015 - model_acc: 0.9886 - val_loss: 0.0140 - val_model_acc: 0.
8377
Epoch 7/20
996/996 [==============================] - 89s 89ms/step - loss: 0.0014 - model_acc: 0.9876 - val_loss: 0.0138 - val_model_acc: 0.
8343
Epoch 8/20
996/996 [==============================] - 90s 91ms/step - loss: 0.0014 - model_acc: 0.9888 - val_loss: 0.0138 - val_model_acc: 0.
8385
Epoch 9/20
996/996 [==============================] - 88s 88ms/step - loss: 0.0013 - model_acc: 0.9879 - val_loss: 0.0139 - val_model_acc: 0.
8343
Epoch 10/20
996/996 [==============================] - 90s 90ms/step - loss: 0.0013 - model_acc: 0.9885 - val_loss: 0.0137 - val_model_acc: 0.
8394
Epoch 11/20
996/996 [==============================] - 83s 83ms/step - loss: 0.0012 - model_acc: 0.9882 - val_loss: 0.0140 - val_model_acc: 0.
8346
Epoch 12/20
996/996 [==============================] - 89s 89ms/step - loss: 0.0012 - model_acc: 0.9899 - val_loss: 0.0139 - val_model_acc: 0.
8366
Epoch 13/20
996/996 [==============================] - 91s 91ms/step - loss: 0.0012 - model_acc: 0.9905 - val_loss: 0.0138 - val_model_acc: 0.
8383
```

```
Epoch 14/20
996/996 [==============================] - 89s 90ms/step - loss: 0.0012 - model_acc: 0.9912 - val_loss: 0.0140 - val_model_acc: 0.
8360
Epoch 15/20
996/996 [==============================] - 88s 89ms/step - loss: 0.0011 - model_acc: 0.9906 - val_loss: 0.0138 - val_model_acc: 0.
8352
Epoch 16/20
996/996 [==============================] - 91s 91ms/step - loss: 0.0011 - model_acc: 0.9900 - val_loss: 0.0140 - val_model_acc: 0.
8374
Epoch 17/20
996/996 [==============================] - 88s 88ms/step - loss: 0.0011 - model_acc: 0.9899 - val_loss: 0.0139 - val_model_acc: 0.
8377
Epoch 18/20
996/996 [==============================] - 89s 90ms/step - loss: 0.0010 - model_acc: 0.9913 - val_loss: 0.0141 - val_model_acc: 0.
8397
Epoch 19/20
996/996 [==============================] - 85s 85ms/step - loss: 0.0010 - model_acc: 0.9918 - val_loss: 0.0141 - val_model_acc: 0.
8377
Epoch 20/20
996/996 [==============================] - 91s 92ms/step - loss: 9.9170e-04 - model_acc: 0.9915 - val_loss: 0.0142 - val_model_ac
c: 0.8422
```

Out[8]:
```
<tensorflow.python.keras.callbacks.History at 0x146a0e1b9f40>
```

In [10]:
```
model.compile(optimizer=adam.Adam(learning_rate=1e-5),
              loss=mse,
              metrics = [model_acc])

model.fit(x=tf.image.grayscale_to_rgb(training_images),
          y=training_emotions,
          batch_size=32,
          epochs=10,
          validation_data=(tf.image.grayscale_to_rgb(test_images), test_emotions))
```

```
Epoch 1/10
996/996 [==============================] - 90s 91ms/step - loss: 9.8101e-04 - model_acc: 0.9910 - val_loss: 0.0142 - val_model_ac
c: 0.8368
Epoch 2/10
996/996 [==============================] - 92s 92ms/step - loss: 9.6246e-04 - model_acc: 0.9918 - val_loss: 0.0140 - val_model_ac
c: 0.8363
Epoch 3/10
996/996 [==============================] - 90s 90ms/step - loss: 9.3209e-04 - model_acc: 0.9931 - val_loss: 0.0141 - val_model_ac
c: 0.8371
```

```
Epoch 4/10
996/996 [==============================] - 90s 90ms/step - loss: 9.2231e-04 - model_acc: 0.9937 - val_loss: 0.0142 - val_model_ac
c: 0.8399
Epoch 5/10
996/996 [==============================] - 91s 91ms/step - loss: 8.9386e-04 - model_acc: 0.9926 - val_loss: 0.0143 - val_model_ac
c: 0.8343
Epoch 6/10
996/996 [==============================] - 89s 89ms/step - loss: 8.9135e-04 - model_acc: 0.9919 - val_loss: 0.0145 - val_model_ac
c: 0.8363
Epoch 7/10
996/996 [==============================] - 91s 91ms/step - loss: 8.5526e-04 - model_acc: 0.9922 - val_loss: 0.0144 - val_model_ac
c: 0.8397
Epoch 8/10
996/996 [==============================] - 89s 90ms/step - loss: 8.5062e-04 - model_acc: 0.9927 - val_loss: 0.0142 - val_model_ac
c: 0.8366
Epoch 9/10
996/996 [==============================] - 73s 73ms/step - loss: 8.5518e-04 - model_acc: 0.9924 - val_loss: 0.0144 - val_model_ac
c: 0.8397
Epoch 10/10
996/996 [==============================] - 90s 91ms/step - loss: 8.3444e-04 - model_acc: 0.9934 - val_loss: 0.0146 - val_model_ac
c: 0.8388
```

Out[10]:
```
<tensorflow.python.keras.callbacks.History at 0x146a202971f0>
```

In [11]:
```
model.compile(optimizer=adam.Adam(learning_rate=1e-5),
              loss=mse,
              metrics = [model_acc])

model.fit(x=tf.image.grayscale_to_rgb(training_images),
          y=training_emotions,
          batch_size=32,
          epochs=10,
          validation_data=(tf.image.grayscale_to_rgb(test_images), test_emotions))
```

```
Epoch 1/10
996/996 [==============================] - 91s 91ms/step - loss: 8.1120e-04 - model_acc: 0.9920 - val_loss: 0.0146 - val_model_ac
c: 0.8394
Epoch 2/10
996/996 [==============================] - 84s 84ms/step - loss: 7.8537e-04 - model_acc: 0.9930 - val_loss: 0.0144 - val_model_ac
c: 0.8399
Epoch 3/10
996/996 [==============================] - 89s 90ms/step - loss: 7.9375e-04 - model_acc: 0.9938 - val_loss: 0.0143 - val_model_ac
c: 0.8394
```

```
Epoch 4/10
996/996 [==============================] - 87s 87ms/step - loss: 7.6594e-04 - model_acc: 0.9919 - val_loss: 0.0143 - val_model_ac
c: 0.8411
Epoch 5/10
996/996 [==============================] - 90s 90ms/step - loss: 7.4503e-04 - model_acc: 0.9937 - val_loss: 0.0146 - val_model_ac
c: 0.8391
Epoch 6/10
996/996 [==============================] - 88s 89ms/step - loss: 7.4686e-04 - model_acc: 0.9939 - val_loss: 0.0146 - val_model_ac
c: 0.8399
Epoch 7/10
996/996 [==============================] - 83s 84ms/step - loss: 7.3835e-04 - model_acc: 0.9927 - val_loss: 0.0146 - val_model_ac
c: 0.8391
Epoch 8/10
996/996 [==============================] - 89s 89ms/step - loss: 7.1994e-04 - model_acc: 0.9936 - val_loss: 0.0143 - val_model_ac
c: 0.8383
Epoch 9/10
996/996 [==============================] - 85s 86ms/step - loss: 7.2759e-04 - model_acc: 0.9926 - val_loss: 0.0145 - val_model_ac
c: 0.8354
Epoch 10/10
996/996 [==============================] - 90s 90ms/step - loss: 6.9369e-04 - model_acc: 0.9929 - val_loss: 0.0146 - val_model_ac
c: 0.8399
```

Out[11]:
```
<tensorflow.python.keras.callbacks.History at 0x1469987d87c0>
```

In [12]:
```
model.compile(optimizer=adam.Adam(learning_rate=1e-5),
              loss=mse,
              metrics = [model_acc])

model.fit(x=tf.image.grayscale_to_rgb(training_images),
          y=training_emotions,
          batch_size=32,
          epochs=10,
          validation_data=(tf.image.grayscale_to_rgb(test_images), test_emotions))
```

```
Epoch 1/10
996/996 [==============================] - 90s 90ms/step - loss: 6.9107e-04 - model_acc: 0.9927 - val_loss: 0.0146 - val_model_ac
c: 0.8377
Epoch 2/10
996/996 [==============================] - 90s 90ms/step - loss: 6.8500e-04 - model_acc: 0.9937 - val_loss: 0.0145 - val_model_ac
c: 0.8374
Epoch 3/10
996/996 [==============================] - 89s 89ms/step - loss: 6.5860e-04 - model_acc: 0.9942 - val_loss: 0.0145 - val_model_ac
c: 0.8377
```

```
Epoch 4/10
996/996 [==============================] - 91s 91ms/step - loss: 6.6356e-04 - model_acc: 0.9942 - val_loss: 0.0145 - val_model_ac
c: 0.8408
Epoch 5/10
996/996 [==============================] - 92s 93ms/step - loss: 6.5697e-04 - model_acc: 0.9943 - val_loss: 0.0145 - val_model_ac
c: 0.8405
Epoch 6/10
996/996 [==============================] - 90s 91ms/step - loss: 6.3380e-04 - model_acc: 0.9943 - val_loss: 0.0146 - val_model_ac
c: 0.8408
Epoch 7/10
996/996 [==============================] - 91s 91ms/step - loss: 6.3052e-04 - model_acc: 0.9935 - val_loss: 0.0145 - val_model_ac
c: 0.8399
Epoch 8/10
996/996 [==============================] - 90s 91ms/step - loss: 6.2349e-04 - model_acc: 0.9939 - val_loss: 0.0145 - val_model_ac
c: 0.8397
Epoch 9/10
996/996 [==============================] - 92s 92ms/step - loss: 6.2658e-04 - model_acc: 0.9938 - val_loss: 0.0146 - val_model_ac
c: 0.8405
Epoch 10/10
996/996 [==============================] - 91s 92ms/step - loss: 6.0977e-04 - model_acc: 0.9944 - val_loss: 0.0147 - val_model_ac
c: 0.8366
```

Out[12]: <tensorflow.python.keras.callbacks.History at 0x1469984cb910>

In [ ]: