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
```

```
2021-12-27 22:49:24.409897: 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)
```

```
2021-12-27 22:49:28.588169: I tensorflow/compiler/jit/xla_cpu_device.cc:41] Not creating XLA devices, tf_xla_enable_xla_devices no
t set
2021-12-27 22:49:28.589686: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcuda.so.1
2021-12-27 22:49:28.630756: 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
2021-12-27 22:49:28.631396: 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
2021-12-27 22:49:28.631431: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcudart.so.11.0
2021-12-27 22:49:28.636260: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcublas.so.11
2021-12-27 22:49:28.636353: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcublasLt.so.11
2021-12-27 22:49:28.638863: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcufft.so.10
2021-12-27 22:49:28.639926: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcurand.so.10
2021-12-27 22:49:28.644906: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcusolver.so.10
2021-12-27 22:49:28.646643: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcusparse.so.11
2021-12-27 22:49:28.647538: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcudnn.so.8
2021-12-27 22:49:28.647702: 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
2021-12-27 22:49:28.648446: 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
2021-12-27 22:49:28.649099: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1862] Adding visible gpu devices: 0
2021-12-27 22:49:28.650631: 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
```

```
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2021-12-27 22:49:28.651432: I tensorflow/compiler/jit/xla_gpu_device.cc:99] Not creating XLA devices, tf_xla_enable_xla_devices no
t set
2021-12-27 22:49:28.651606: 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
2021-12-27 22:49:28.652465: 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
2021-12-27 22:49:28.652515: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcudart.so.11.0
2021-12-27 22:49:28.652538: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcublas.so.11
2021-12-27 22:49:28.652553: I tensorflow/stream_executor/platform/de
```

In [5]:

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

```
fault/dso_loader.cc:49] Successfully opened dynamic library libcublasLt.so.11
2021-12-27 22:49:28.652569: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcufft.so.10
2021-12-27 22:49:28.652584: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcurand.so.10
2021-12-27 22:49:28.652715: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcusolver.so.10
2021-12-27 22:49:28.652733: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcusparse.so.11
2021-12-27 22:49:28.652749: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcudnn.so.8
2021-12-27 22:49:28.652839: 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
2021-12-27 22:49:28.653657: 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
2021-12-27 22:49:28.654312: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1862] Adding visible gpu devices: 0
2021-12-27 22:49:28.654370: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcudart.so.11.0
2021-12-27 22:49:29.577262: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1261] Device interconnect StreamExecutor with stren
gth 1 edge matrix:
2021-12-27 22:49:29.577301: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1267]      0
2021-12-27 22:49:29.577309: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1280] 0:   N
2021-12-27 22:49:29.577508: 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
2021-12-27 22:49:29.578026: 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
2021-12-27 22:49:29.578504: 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
2021-12-27 22:49:29.578921: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1406] Created TensorFlow device (/job:localhost/rep
lica:0/task:0/device:GPU:0 with 10071 MB memory) -> physical GPU (device: 0, name: GeForce RTX 2080 Ti, pci bus id: 0000:05:00.0,
compute capability: 7.5)
```

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 = Flatten()(x)

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.trainable=True
```

```
model.summary()

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=30,
          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] 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]

```
_________________________________________________________________________________________________
dense_1 (Dense)                 (None, 4096)         16781312    dense[0][0]
_________________________________________________________________________________________________
dense_2 (Dense)                 (None, 8)            32776       dense_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(
2021-12-27 22:49:31.673604: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:116] None of the MLIR optimization passes a
re enabled (registered 2)
2021-12-27 22:49:31.674472: I tensorflow/core/platform/profile_utils/cpu_utils.cc:112] CPU Frequency: 2199995000 Hz
2021-12-27 22:49:31.698172: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcudnn.so.8
Epoch 1/30
2021-12-27 22:49:33.902236: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcublas.so.11
2021-12-27 22:49:34.527928: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library li
bcublasLt.so.11
996/996 [==============================] - 96s 93ms/step - loss: 0.0514 - model_acc: 0.5094 - val_loss: 0.0361 - val_model_acc: 0.
6397
Epoch 2/30
996/996 [==============================] - 91s 91ms/step - loss: 0.0287 - model_acc: 0.6933 - val_loss: 0.0260 - val_model_acc: 0.
7129
Epoch 3/30
996/996 [==============================] - 82s 82ms/step - loss: 0.0201 - model_acc: 0.7696 - val_loss: 0.0205 - val_model_acc: 0.
7611
Epoch 4/30
996/996 [==============================] - 89s 89ms/step - loss: 0.0163 - model_acc: 0.8037 - val_loss: 0.0214 - val_model_acc: 0.
7608
Epoch 5/30
996/996 [==============================] - 86s 87ms/step - loss: 0.0137 - model_acc: 0.8333 - val_loss: 0.0204 - val_model_acc: 0.
7668
Epoch 6/30
119/996 [==>...........................] - ETA: 1:15 - loss: 0.0127 - model_acc: 0.8461
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
/tmp/.fym666/ipykernel_26051/354638722.py in <module>
     97                 metrics = [model_acc])
```

```
  10443         begin_mask, "end_mask", end_mask, "ellipsis_mask", ellipsis_mask,

KeyboardInterrupt:
```

In [9]:

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

model.fit(x=tf.image.grayscale_to_rgb(training_images),
          y=training_emotions,
          batch_size=32,
          epochs=30,
          validation_data=(tf.image.grayscale_to_rgb(test_images), test_emotions))
```

```
Epoch 1/30
996/996 [==============================] - 83s 84ms/step - loss: 0.0270 - model_acc: 0.7267 - val_loss: 0.0205 - val_model_acc: 0.
7718
Epoch 2/30
996/996 [==============================] - 89s 89ms/step - loss: 0.0151 - model_acc: 0.8210 - val_loss: 0.0176 - val_model_acc: 0.
7913
Epoch 3/30
996/996 [==============================] - 90s 90ms/step - loss: 0.0120 - model_acc: 0.8554 - val_loss: 0.0160 - val_model_acc: 0.
8098
Epoch 4/30
996/996 [==============================] - 88s 89ms/step - loss: 0.0101 - model_acc: 0.8726 - val_loss: 0.0151 - val_model_acc: 0.
8189
Epoch 5/30
996/996 [==============================] - 88s 88ms/step - loss: 0.0085 - model_acc: 0.8897 - val_loss: 0.0161 - val_model_acc: 0.
8160
Epoch 6/30
996/996 [==============================] - 91s 92ms/step - loss: 0.0072 - model_acc: 0.9082 - val_loss: 0.0148 - val_model_acc: 0.
8242
Epoch 7/30
996/996 [==============================] - 87s 87ms/step - loss: 0.0063 - model_acc: 0.9228 - val_loss: 0.0156 - val_model_acc: 0.
8152
Epoch 8/30
996/996 [==============================] - 90s 91ms/step - loss: 0.0057 - model_acc: 0.9276 - val_loss: 0.0141 - val_model_acc: 0.
8245
Epoch 9/30
996/996 [==============================] - 90s 90ms/step - loss: 0.0051 - model_acc: 0.9348 - val_loss: 0.0144 - val_model_acc: 0.
8230
Epoch 10/30
996/996 [==============================] - 85s 85ms/step - loss: 0.0046 - model_acc: 0.9420 - val_loss: 0.0147 - val_model_acc: 0.
```

```
8250
Epoch 11/30
996/996 [==============================] - 90s 90ms/step - loss: 0.0041 - model_acc: 0.9472 - val_loss: 0.0134 - val_model_acc: 0.
8364
Epoch 12/30
996/996 [==============================] - 90s 90ms/step - loss: 0.0039 - model_acc: 0.9496 - val_loss: 0.0140 - val_model_acc: 0.
8296
Epoch 13/30
996/996 [==============================] - 88s 89ms/step - loss: 0.0034 - model_acc: 0.9560 - val_loss: 0.0137 - val_model_acc: 0.
8380
Epoch 14/30
996/996 [==============================] - 75s 76ms/step - loss: 0.0032 - model_acc: 0.9578 - val_loss: 0.0132 - val_model_acc: 0.
8343
Epoch 15/30
996/996 [==============================] - 83s 83ms/step - loss: 0.0030 - model_acc: 0.9581 - val_loss: 0.0133 - val_model_acc: 0.
8323
Epoch 16/30
996/996 [==============================] - 88s 88ms/step - loss: 0.0028 - model_acc: 0.9613 - val_loss: 0.0132 - val_model_acc: 0.
8346
Epoch 17/30
996/996 [==============================] - 79s 79ms/step - loss: 0.0026 - model_acc: 0.9633 - val_loss: 0.0134 - val_model_acc: 0.
8374
Epoch 18/30
996/996 [==============================] - 91s 92ms/step - loss: 0.0024 - model_acc: 0.9671 - val_loss: 0.0132 - val_model_acc: 0.
8374
Epoch 19/30
996/996 [==============================] - 86s 87ms/step - loss: 0.0022 - model_acc: 0.9702 - val_loss: 0.0131 - val_model_acc: 0.
8363
Epoch 20/30
996/996 [==============================] - 82s 82ms/step - loss: 0.0021 - model_acc: 0.9715 - val_loss: 0.0129 - val_model_acc: 0.
8405
Epoch 21/30
996/996 [==============================] - 84s 85ms/step - loss: 0.0021 - model_acc: 0.9707 - val_loss: 0.0129 - val_model_acc: 0.
8425
Epoch 22/30
996/996 [==============================] - 81s 81ms/step - loss: 0.0020 - model_acc: 0.9710 - val_loss: 0.0131 - val_model_acc: 0.
8363
Epoch 23/30
996/996 [==============================] - 90s 90ms/step - loss: 0.0017 - model_acc: 0.9748 - val_loss: 0.0127 - val_model_acc: 0.
8428
Epoch 24/30
996/996 [==============================] - 89s 90ms/step - loss: 0.0016 - model_acc: 0.9775 - val_loss: 0.0129 - val_model_acc: 0.
8399
Epoch 25/30
```

```
996/996 [==============================] - 91s 91ms/step - loss: 0.0016 - model_acc: 0.9760 - val_loss: 0.0128 - val_model_acc: 0.
8447
Epoch 26/30
996/996 [==============================] - 86s 87ms/step - loss: 0.0016 - model_acc: 0.9751 - val_loss: 0.0127 - val_model_acc: 0.
8445
Epoch 27/30
996/996 [==============================] - 90s 90ms/step - loss: 0.0015 - model_acc: 0.9787 - val_loss: 0.0127 - val_model_acc: 0.
8416
Epoch 28/30
996/996 [==============================] - 80s 80ms/step - loss: 0.0013 - model_acc: 0.9811 - val_loss: 0.0127 - val_model_acc: 0.
8438
Epoch 29/30
996/996 [==============================] - 76s 76ms/step - loss: 0.0013 - model_acc: 0.9807 - val_loss: 0.0127 - val_model_acc: 0.
8428
Epoch 30/30
996/996 [==============================] - 91s 91ms/step - loss: 0.0012 - model_acc: 0.9828 - val_loss: 0.0130 - val_model_acc: 0.
8402
```

Out[9]:
```
<tensorflow.python.keras.callbacks.History at 0x14ca1d005640>
```

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 [==============================] - 88s 89ms/step - loss: 9.2002e-04 - model_acc: 0.9883 - val_loss: 0.0121 - val_model_ac
c: 0.8517
Epoch 2/10
996/996 [==============================] - 86s 87ms/step - loss: 6.0451e-04 - model_acc: 0.9973 - val_loss: 0.0121 - val_model_ac
c: 0.8517
Epoch 3/10
996/996 [==============================] - 90s 91ms/step - loss: 4.9480e-04 - model_acc: 0.9984 - val_loss: 0.0121 - val_model_ac
c: 0.8520
Epoch 4/10
996/996 [==============================] - 91s 92ms/step - loss: 4.3324e-04 - model_acc: 0.9986 - val_loss: 0.0121 - val_model_ac
c: 0.8506
Epoch 5/10
```

```
996/996 [==============================] - 85s 85ms/step - loss: 3.9408e-04 - model_acc: 0.9988 - val_loss: 0.0121 - val_model_ac
c: 0.8509
Epoch 6/10
996/996 [==============================] - 81s 82ms/step - loss: 3.5657e-04 - model_acc: 0.9989 - val_loss: 0.0122 - val_model_ac
c: 0.8512
Epoch 7/10
996/996 [==============================] - 90s 91ms/step - loss: 3.3375e-04 - model_acc: 0.9991 - val_loss: 0.0122 - val_model_ac
c: 0.8517
Epoch 8/10
996/996 [==============================] - 90s 91ms/step - loss: 3.1838e-04 - model_acc: 0.9992 - val_loss: 0.0123 - val_model_ac
c: 0.8504
Epoch 9/10
996/996 [==============================] - 89s 89ms/step - loss: 3.0083e-04 - model_acc: 0.9988 - val_loss: 0.0122 - val_model_ac
c: 0.8510
Epoch 10/10
996/996 [==============================] - 88s 89ms/step - loss: 2.7178e-04 - model_acc: 0.9989 - val_loss: 0.0123 - val_model_ac
c: 0.8501
```

Out[10]: `<tensorflow.python.keras.callbacks.History at 0x14c9a14190a0>`

In [ ]: