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

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
if tf.test.gpu_device_name():
    print('GPU found')
else:
    print("No GPU found")
```

```
GPU found
```

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)
```

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
from tensorflow.python.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.python.keras.models import Model
from tensorflow.python.keras import layers, Sequential

base_model = vgg16.VGG16(include_top=False, weights="imagenet", input_shape=(48,48,3))
base_model.trainable=True
model = Sequential([
    base_model,
    layers.GlobalAveragePooling2D(),
    layers.Dense(2048, activation='relu'),
    layers.Dense(2048, activation='relu'),
    layers.Dense(emotions_count, activation='softmax'),
])
model.compile(optimizer=adam.Adam(learning_rate=1e-4), loss=mse, metrics = [model_acc])
model.fit(x=tf.image.grayscale_to_rgb(training_images),
          y=training_emotions,
          batch_size=32,
          epochs=40,
          validation_data=(tf.image.grayscale_to_rgb(test_images), test_emotions))
#model.summary()
#base_model = resnet.ResNet50(include_top=False, weights="imagenet", input_shape=(48,48,3))
```

```
C:\Users\Darkl\anaconda3\lib\site-packages\tensorflow\python\data\ops\dataset_ops.py:3703: UserWarning: Even though the `
tf.config.experimental_run_functions_eagerly` option is set, this option does not apply to tf.data functions. To force ea
ger execution of tf.data functions, please use `tf.data.experimental.enable.debug_mode()`.
  warnings.warn(
Epoch 1/40
996/996 [==============================] - 73s 70ms/step - loss: 0.0299 - model_acc: 0.6776 - val_loss: 0.0230 - val_mode
l_acc: 0.7382
Epoch 2/40
996/996 [==============================] - 69s 69ms/step - loss: 0.0186 - model_acc: 0.7817 - val_loss: 0.0185 - val_mode
l_acc: 0.7771
Epoch 3/40
996/996 [==============================] - 69s 70ms/step - loss: 0.0144 - model_acc: 0.8230 - val_loss: 0.0174 - val_mode
l_acc: 0.7943
Epoch 4/40
996/996 [==============================] - 68s 68ms/step - loss: 0.0116 - model_acc: 0.8544 - val_loss: 0.0171 - val_mode
```

```
l_acc: 0.7912
Epoch 5/40
996/996 [==============================] - 67s 67ms/step - loss: 0.0096 - model_acc: 0.8778 - val_loss: 0.0155 - val_mode
l_acc: 0.8134
Epoch 6/40
996/996 [==============================] - 67s 67ms/step - loss: 0.0081 - model_acc: 0.8977 - val_loss: 0.0143 - val_mode
l_acc: 0.8225
Epoch 7/40
996/996 [==============================] - 67s 68ms/step - loss: 0.0068 - model_acc: 0.9121 - val_loss: 0.0144 - val_mode
l_acc: 0.8275
Epoch 8/40
996/996 [==============================] - 67s 67ms/step - loss: 0.0057 - model_acc: 0.9271 - val_loss: 0.0144 - val_mode
l_acc: 0.8264
Epoch 9/40
996/996 [==============================] - 67s 67ms/step - loss: 0.0053 - model_acc: 0.9324 - val_loss: 0.0135 - val_mode
l_acc: 0.8294
Epoch 10/40
996/996 [==============================] - 67s 67ms/step - loss: 0.0044 - model_acc: 0.9425 - val_loss: 0.0141 - val_mode
l_acc: 0.8230
Epoch 11/40
996/996 [==============================] - 67s 67ms/step - loss: 0.0041 - model_acc: 0.9463 - val_loss: 0.0142 - val_mode
l_acc: 0.8230
Epoch 12/40
996/996 [==============================] - 67s 67ms/step - loss: 0.0036 - model_acc: 0.9536 - val_loss: 0.0131 - val_mode
l_acc: 0.8363
Epoch 13/40
996/996 [==============================] - 69s 69ms/step - loss: 0.0032 - model_acc: 0.9564 - val_loss: 0.0135 - val_mode
l_acc: 0.8346
Epoch 14/40
996/996 [==============================] - 68s 68ms/step - loss: 0.0030 - model_acc: 0.9610 - val_loss: 0.0130 - val_mode
l_acc: 0.8380
Epoch 15/40
996/996 [==============================] - 68s 68ms/step - loss: 0.0029 - model_acc: 0.9610 - val_loss: 0.0131 - val_mode
l_acc: 0.8331
Epoch 16/40
996/996 [==============================] - 68s 68ms/step - loss: 0.0025 - model_acc: 0.9660 - val_loss: 0.0126 - val_mode
l_acc: 0.8453
Epoch 17/40
996/996 [==============================] - 67s 68ms/step - loss: 0.0025 - model_acc: 0.9633 - val_loss: 0.0127 - val_mode
l_acc: 0.8402
Epoch 18/40
996/996 [==============================] - 69s 70ms/step - loss: 0.0022 - model_acc: 0.9689 - val_loss: 0.0128 - val_mode
l_acc: 0.8399
Epoch 19/40
996/996 [==============================] - 68s 68ms/step - loss: 0.0020 - model_acc: 0.9721 - val_loss: 0.0129 - val_mode
l_acc: 0.8399
```

```
Epoch 20/40
996/996 [==============================] - 68s 68ms/step - loss: 0.0019 - model_acc: 0.9739 - val_loss: 0.0127 - val_mode
l_acc: 0.8365
Epoch 21/40
996/996 [==============================] - 68s 69ms/step - loss: 0.0020 - model_acc: 0.9710 - val_loss: 0.0126 - val_mode
l_acc: 0.8413
Epoch 22/40
996/996 [==============================] - 70s 70ms/step - loss: 0.0017 - model_acc: 0.9747 - val_loss: 0.0125 - val_mode
l_acc: 0.8428
Epoch 23/40
996/996 [==============================] - 70s 70ms/step - loss: 0.0015 - model_acc: 0.9787 - val_loss: 0.0128 - val_mode
l_acc: 0.8405
Epoch 24/40
996/996 [==============================] - 70s 70ms/step - loss: 0.0021 - model_acc: 0.9687 - val_loss: 0.0129 - val_mode
l_acc: 0.8365
Epoch 25/40
996/996 [==============================] - 71s 71ms/step - loss: 0.0015 - model_acc: 0.9793 - val_loss: 0.0125 - val_mode
l_acc: 0.8498
Epoch 26/40
996/996 [==============================] - 71s 71ms/step - loss: 0.0011 - model_acc: 0.9878 - val_loss: 0.0125 - val_mode
l_acc: 0.8427
Epoch 27/40
996/996 [==============================] - 69s 70ms/step - loss: 0.0011 - model_acc: 0.9860 - val_loss: 0.0125 - val_mode
l_acc: 0.8458
Epoch 28/40
996/996 [==============================] - 68s 69ms/step - loss: 0.0012 - model_acc: 0.9820 - val_loss: 0.0125 - val_mode
l_acc: 0.8464
Epoch 29/40
996/996 [==============================] - 70s 70ms/step - loss: 0.0012 - model_acc: 0.9821 - val_loss: 0.0129 - val_mode
l_acc: 0.8368
Epoch 30/40
996/996 [==============================] - 69s 69ms/step - loss: 0.0011 - model_acc: 0.9827 - val_loss: 0.0125 - val_mode
l_acc: 0.8467
Epoch 31/40
996/996 [==============================] - 69s 70ms/step - loss: 0.0011 - model_acc: 0.9836 - val_loss: 0.0127 - val_mode
l_acc: 0.8416
Epoch 32/40
996/996 [==============================] - 70s 70ms/step - loss: 0.0015 - model_acc: 0.9762 - val_loss: 0.0151 - val_mode
l_acc: 0.8250
Epoch 33/40
996/996 [==============================] - 70s 71ms/step - loss: 0.0013 - model_acc: 0.9796 - val_loss: 0.0128 - val_mode
l_acc: 0.8389
Epoch 34/40
996/996 [==============================] - 70s 70ms/step - loss: 7.1950e-04 - model_acc: 0.9917 - val_loss: 0.0125 - val_
model_acc: 0.8461
Epoch 35/40
```

```
996/996 [==============================] - 71s 71ms/step - loss: 6.0392e-04 - model_acc: 0.9941 - val_loss: 0.0126 - val_
model_acc: 0.8465
Epoch 36/40
996/996 [==============================] - 69s 69ms/step - loss: 6.9085e-04 - model_acc: 0.9899 - val_loss: 0.0129 - val_
model_acc: 0.8436
Epoch 37/40
996/996 [==============================] - 71s 71ms/step - loss: 8.3781e-04 - model_acc: 0.9871 - val_loss: 0.0131 - val_
model_acc: 0.8351
Epoch 38/40
996/996 [==============================] - 71s 71ms/step - loss: 9.0675e-04 - model_acc: 0.9852 - val_loss: 0.0127 - val_
model_acc: 0.8444
Epoch 39/40
996/996 [==============================] - 68s 69ms/step - loss: 7.2848e-04 - model_acc: 0.9878 - val_loss: 0.0124 - val_
model_acc: 0.8487
Epoch 40/40
996/996 [==============================] - 70s 71ms/step - loss: 6.7742e-04 - model_acc: 0.9900 - val_loss: 0.0126 - val_
model_acc: 0.8438
```

Out[8]: <tensorflow.python.keras.callbacks.History at 0x20c9cb47940>

In [ ]: