In [9]:

```
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.python.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.python.keras.models import Model
from tensorflow.python.keras import layers, Sequential,losses, metrics

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)

image_path = "./dataset/images.npy"
emotion_multi_path = "./dataset/emotions_multi.npy"
emotion_single_path = "./dataset/emotions_single.npy"
```

In [10]:

```
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 [11]:

```
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 [12]:

```
#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)
# images = tf.image.resize(images, [224,224])
images = layers.Rescaling(1./127.5, offset= -1)(images)

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, 3)
training_emotions shape: (31858, 8)
test_images shape: (3535, 48, 48, 3)
test_emotions shape: (3535, 8)
```

In [13]:

```
from tensorflow.python.keras.applications import vgg16, resnet
from tensorflow.python.keras import optimizers
from tensorflow.python.keras.optimizer_v2 import adam
```

In [14]:

```
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=losses.CategoricalCrossentropy(),
              metrics = [model_acc])

model.fit(x=training_images,
          y=training_emotions,
          batch_size=32,
          epochs=40,
          validation_data=(test_images, test_emotions))
```

```
Epoch 1/40
996/996 [==============================] - 69s 69ms/step - loss: 1.1220 - model_acc: 0.7052 - val_loss: 0.9490 - val_model_ac
c: 0.7726
Epoch 2/40
996/996 [==============================] - 69s 69ms/step - loss: 0.9274 - model_acc: 0.7986 - val_loss: 0.9093 - val_model_ac
c: 0.7822
Epoch 3/40
996/996 [==============================] - 68s 68ms/step - loss: 0.8545 - model_acc: 0.8396 - val_loss: 0.8735 - val_model_ac
c: 0.8084
Epoch 4/40
996/996 [==============================] - 68s 68ms/step - loss: 0.8011 - model_acc: 0.8706 - val_loss: 0.8622 - val_model_ac
c: 0.8245
Epoch 5/40
996/996 [==============================] - 68s 68ms/step - loss: 0.7665 - model_acc: 0.8918 - val_loss: 0.8751 - val_model_ac
c: 0.8222
Epoch 6/40
996/996 [==============================] - 67s 67ms/step - loss: 0.7389 - model_acc: 0.9074 - val_loss: 0.8504 - val_model_ac
c: 0.8309
Epoch 7/40
996/996 [==============================] - 67s 68ms/step - loss: 0.7126 - model_acc: 0.9202 - val_loss: 0.8663 - val_model_ac
c: 0.8264
Epoch 8/40
996/996 [==============================] - 67s 67ms/step - loss: 0.6908 - model_acc: 0.9295 - val_loss: 0.8614 - val_model_ac
c: 0.8323
Epoch 9/40
996/996 [==============================] - 67s 68ms/step - loss: 0.6703 - model_acc: 0.9360 - val_loss: 0.8779 - val_model_ac
```

```
c: 0.8278
Epoch 10/40
996/996 [==============================] - 67s 68ms/step - loss: 0.6537 - model_acc: 0.9358 - val_loss: 0.8666 - val_model_ac
c: 0.8395
Epoch 11/40
996/996 [==============================] - 68s 68ms/step - loss: 0.6344 - model_acc: 0.9449 - val_loss: 0.8821 - val_model_ac
c: 0.8337
Epoch 12/40
996/996 [==============================] - 67s 67ms/step - loss: 0.6181 - model_acc: 0.9477 - val_loss: 0.9078 - val_model_ac
c: 0.8340
Epoch 13/40
996/996 [==============================] - 68s 68ms/step - loss: 0.6044 - model_acc: 0.9505 - val_loss: 0.9286 - val_model_ac
c: 0.8306
Epoch 14/40
996/996 [==============================] - 70s 70ms/step - loss: 0.5952 - model_acc: 0.9504 - val_loss: 0.9544 - val_model_ac
c: 0.8364
Epoch 15/40
996/996 [==============================] - 68s 68ms/step - loss: 0.5865 - model_acc: 0.9543 - val_loss: 0.9604 - val_model_ac
c: 0.8400
Epoch 16/40
996/996 [==============================] - 67s 68ms/step - loss: 0.5810 - model_acc: 0.9576 - val_loss: 0.9478 - val_model_ac
c: 0.8389
Epoch 17/40
996/996 [==============================] - 67s 67ms/step - loss: 0.5769 - model_acc: 0.9591 - val_loss: 0.9905 - val_model_ac
c: 0.8405
Epoch 18/40
996/996 [==============================] - 67s 67ms/step - loss: 0.5742 - model_acc: 0.9571 - val_loss: 0.9834 - val_model_ac
c: 0.8360
Epoch 19/40
996/996 [==============================] - 68s 68ms/step - loss: 0.5711 - model_acc: 0.9600 - val_loss: 1.0250 - val_model_ac
c: 0.8450
Epoch 20/40
996/996 [==============================] - 69s 69ms/step - loss: 0.5688 - model_acc: 0.9619 - val_loss: 1.0216 - val_model_ac
c: 0.8374
Epoch 21/40
996/996 [==============================] - 68s 69ms/step - loss: 0.5679 - model_acc: 0.9622 - val_loss: 1.0434 - val_model_ac
c: 0.8284
Epoch 22/40
996/996 [==============================] - 67s 67ms/step - loss: 0.5650 - model_acc: 0.9642 - val_loss: 1.0283 - val_model_ac
c: 0.8352
Epoch 23/40
996/996 [==============================] - 67s 67ms/step - loss: 0.5645 - model_acc: 0.9636 - val_loss: 1.0279 - val_model_ac
c: 0.8371
Epoch 24/40
```

```
996/996 [==============================] - 67s 68ms/step - loss: 0.5631 - model_acc: 0.9652 - val_loss: 1.0562 - val_model_ac
c: 0.8354
Epoch 25/40
996/996 [==============================] - 67s 68ms/step - loss: 0.5607 - model_acc: 0.9667 - val_loss: 1.0597 - val_model_ac
c: 0.8388
Epoch 26/40
996/996 [==============================] - 67s 67ms/step - loss: 0.5610 - model_acc: 0.9646 - val_loss: 1.0629 - val_model_ac
c: 0.8304
Epoch 27/40
996/996 [==============================] - 67s 67ms/step - loss: 0.5591 - model_acc: 0.9680 - val_loss: 1.0739 - val_model_ac
c: 0.8374
Epoch 28/40
996/996 [==============================] - 69s 69ms/step - loss: 0.5587 - model_acc: 0.9674 - val_loss: 1.0534 - val_model_ac
c: 0.8382
Epoch 29/40
996/996 [==============================] - 68s 68ms/step - loss: 0.5596 - model_acc: 0.9661 - val_loss: 1.0743 - val_model_ac
c: 0.82980.559
Epoch 30/40
996/996 [==============================] - 68s 68ms/step - loss: 0.5564 - model_acc: 0.9703 - val_loss: 1.0857 - val_model_ac
c: 0.8309
Epoch 31/40
996/996 [==============================] - 68s 68ms/step - loss: 0.5567 - model_acc: 0.9688 - val_loss: 1.1141 - val_model_ac
c: 0.8380
Epoch 32/40
996/996 [==============================] - 70s 70ms/step - loss: 0.5561 - model_acc: 0.9697 - val_loss: 1.0710 - val_model_ac
c: 0.8347
Epoch 33/40
996/996 [==============================] - 68s 68ms/step - loss: 0.5554 - model_acc: 0.9704 - val_loss: 1.0975 - val_model_ac
c: 0.8385
Epoch 34/40
996/996 [==============================] - 67s 67ms/step - loss: 0.5538 - model_acc: 0.9715 - val_loss: 1.0677 - val_model_ac
c: 0.8397
Epoch 35/40
996/996 [==============================] - 68s 68ms/step - loss: 0.5537 - model_acc: 0.9706 - val_loss: 1.0982 - val_model_ac
c: 0.8391
Epoch 36/40
996/996 [==============================] - 67s 67ms/step - loss: 0.5535 - model_acc: 0.9727 - val_loss: 1.1157 - val_model_ac
c: 0.8383
Epoch 37/40
996/996 [==============================] - 66s 67ms/step - loss: 0.5521 - model_acc: 0.9737 - val_loss: 1.0997 - val_model_ac
c: 0.8365
Epoch 38/40
996/996 [==============================] - 67s 68ms/step - loss: 0.5538 - model_acc: 0.9710 - val_loss: 1.0971 - val_model_ac
c: 0.8395
```

```
Epoch 39/40
996/996 [==============================] - 66s 67ms/step - loss: 0.5512 - model_acc: 0.9738 - val_loss: 1.1182 - val_model_ac
c: 0.8374
Epoch 40/40
996/996 [==============================] - 68s 69ms/step - loss: 0.5505 - model_acc: 0.9751 - val_loss: 1.0912 - val_model_ac
c: 0.8343
```

Out[14]:

```
<tensorflow.python.keras.callbacks.History at 0x26358974b20>
```

In [17]:

```
base_model = resnet.ResNet101(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=losses.CategoricalCrossentropy(),
              metrics = [model_acc])

model.fit(x=training_images,
          y=training_emotions,
          batch_size=32,
          epochs=40,
          validation_data=(test_images, test_emotions))
```

```
Epoch 1/40
 17/996 [..............................] - ETA: 4:04 - loss: 1.8575 - model_acc: 0.3695
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_4444/1551714665.py in <module>
     15               metrics = [model_acc])
     16 
---> 17 model.fit(x=training_images,
     18           y=training_emotions,
     19           batch_size=32,

~\anaconda3\lib\site-packages\tensorflow\python\keras\engine\training.py in fit(self, x, y, batch_size, epochs, verbose, call
backs, validation_split, validation_data, shuffle, class_weight, sample_weight, initial_epoch, steps_per_epoch, validation_st
```