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, 3)
emotions shape: (35393, 8)
```

In [5]:

```
# images = tf.image.resize(images, [224,224])
# print("images shape:", images.shape)
```

In [6]:

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

In [7]:

```
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 [8]:

```
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 [9]:

```
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=training_images,
          y=training_emotions,
          batch_size=32,
          epochs=40,
          validation_data=(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 [==============================] - 71s 68ms/step - loss: 0.0327 - model_acc: 0.6473 - val_loss: 0.0232 - val_mode
l_acc: 0.7357
Epoch 2/40
996/996 [==============================] - 70s 70ms/step - loss: 0.0193 - model_acc: 0.7714 - val_loss: 0.0188 - val_mode
l_acc: 0.7728
Epoch 3/40
```

```
996/996 [==============================] - 68s 69ms/step - loss: 0.0149 - model_acc: 0.8183 - val_loss: 0.0164 - val_mode
l_acc: 0.8008
Epoch 4/40
996/996 [==============================] - 68s 68ms/step - loss: 0.0118 - model_acc: 0.8513 - val_loss: 0.0162 - val_mode
l_acc: 0.8078
Epoch 5/40
996/996 [==============================] - 67s 68ms/step - loss: 0.0099 - model_acc: 0.8738 - val_loss: 0.0161 - val_mode
l_acc: 0.8042
Epoch 6/40
996/996 [==============================] - 68s 68ms/step - loss: 0.0082 - model_acc: 0.8926 - val_loss: 0.0146 - val_mode
l_acc: 0.8255
Epoch 7/40
996/996 [==============================] - 68s 68ms/step - loss: 0.0069 - model_acc: 0.9129 - val_loss: 0.0144 - val_mode
l_acc: 0.8214
Epoch 8/40
996/996 [==============================] - 67s 68ms/step - loss: 0.0060 - model_acc: 0.9227 - val_loss: 0.0151 - val_mode
l_acc: 0.8152
Epoch 9/40
996/996 [==============================] - 67s 67ms/step - loss: 0.0055 - model_acc: 0.9297 - val_loss: 0.0142 - val_mode
l_acc: 0.8239
Epoch 10/40
996/996 [==============================] - 67s 67ms/step - loss: 0.0048 - model_acc: 0.9397 - val_loss: 0.0138 - val_mode
l_acc: 0.8250
Epoch 11/40
996/996 [==============================] - 67s 67ms/step - loss: 0.0041 - model_acc: 0.9484 - val_loss: 0.0150 - val_mode
l_acc: 0.8210
Epoch 12/40
996/996 [==============================] - 68s 68ms/step - loss: 0.0038 - model_acc: 0.9496 - val_loss: 0.0148 - val_mode
l_acc: 0.8224
Epoch 13/40
996/996 [==============================] - 69s 69ms/step - loss: 0.0034 - model_acc: 0.9544 - val_loss: 0.0138 - val_mode
l_acc: 0.8267
Epoch 14/40
996/996 [==============================] - 68s 68ms/step - loss: 0.0031 - model_acc: 0.9598 - val_loss: 0.0134 - val_mode
l_acc: 0.8368
Epoch 15/40
996/996 [==============================] - 68s 69ms/step - loss: 0.0029 - model_acc: 0.9609 - val_loss: 0.0139 - val_mode
l_acc: 0.8346
Epoch 16/40
996/996 [==============================] - 69s 69ms/step - loss: 0.0027 - model_acc: 0.9628 - val_loss: 0.0132 - val_mode
l_acc: 0.8362
Epoch 17/40
996/996 [==============================] - 70s 70ms/step - loss: 0.0025 - model_acc: 0.9655 - val_loss: 0.0140 - val_mode
l_acc: 0.8317
Epoch 18/40
996/996 [==============================] - 68s 68ms/step - loss: 0.0030 - model_acc: 0.9593 - val_loss: 0.0130 - val_mode
```

```
l_acc: 0.8362
Epoch 19/40
996/996 [==============================] - 69s 69ms/step - loss: 0.0021 - model_acc: 0.9718 - val_loss: 0.0124 - val_mode
l_acc: 0.8483
Epoch 20/40
996/996 [==============================] - 68s 68ms/step - loss: 0.0018 - model_acc: 0.9773 - val_loss: 0.0126 - val_mode
l_acc: 0.8405
Epoch 21/40
996/996 [==============================] - 67s 67ms/step - loss: 0.0018 - model_acc: 0.9763 - val_loss: 0.0127 - val_mode
l_acc: 0.8416
Epoch 22/40
996/996 [==============================] - 67s 67ms/step - loss: 0.0019 - model_acc: 0.9729 - val_loss: 0.0131 - val_mode
l_acc: 0.8408
Epoch 23/40
996/996 [==============================] - 68s 68ms/step - loss: 0.0018 - model_acc: 0.9738 - val_loss: 0.0131 - val_mode
l_acc: 0.8422
Epoch 24/40
996/996 [==============================] - 68s 68ms/step - loss: 0.0016 - model_acc: 0.9765 - val_loss: 0.0126 - val_mode
l_acc: 0.8393
Epoch 25/40
996/996 [==============================] - 69s 69ms/step - loss: 0.0015 - model_acc: 0.9789 - val_loss: 0.0129 - val_mode
l_acc: 0.8354
Epoch 26/40
996/996 [==============================] - 69s 69ms/step - loss: 0.0017 - model_acc: 0.9769 - val_loss: 0.0128 - val_mode
l_acc: 0.8428
Epoch 27/40
996/996 [==============================] - 66s 66ms/step - loss: 0.0013 - model_acc: 0.9810 - val_loss: 0.0127 - val_mode
l_acc: 0.8416
Epoch 28/40
996/996 [==============================] - 67s 67ms/step - loss: 0.0011 - model_acc: 0.9848 - val_loss: 0.0126 - val_mode
l_acc: 0.8436
Epoch 29/40
996/996 [==============================] - 66s 66ms/step - loss: 0.0011 - model_acc: 0.9844 - val_loss: 0.0128 - val_mode
l_acc: 0.8377
Epoch 30/40
996/996 [==============================] - 67s 68ms/step - loss: 0.0021 - model_acc: 0.9686 - val_loss: 0.0133 - val_mode
l_acc: 0.8354
Epoch 31/40
996/996 [==============================] - 69s 69ms/step - loss: 0.0011 - model_acc: 0.9851 - val_loss: 0.0124 - val_mode
l_acc: 0.8436
Epoch 32/40
996/996 [==============================] - 67s 68ms/step - loss: 7.3972e-04 - model_acc: 0.9936 - val_loss: 0.0123 - val_
model_acc: 0.8484
Epoch 33/40
996/996 [==============================] - 65s 66ms/step - loss: 7.5954e-04 - model_acc: 0.9915 - val_loss: 0.0130 - val_
model_acc: 0.8407
```

```
Epoch 34/40
996/996 [==============================] - 65s 66ms/step - loss: 9.4160e-04 - model_acc: 0.9865 - val_loss: 0.0128 - val_
model_acc: 0.8453
Epoch 35/40
996/996 [==============================] - 65s 66ms/step - loss: 0.0010 - model_acc: 0.9827 - val_loss: 0.0132 - val_mode
l_acc: 0.8391
Epoch 36/40
996/996 [==============================] - 67s 67ms/step - loss: 0.0014 - model_acc: 0.9777 - val_loss: 0.0150 - val_mode
l_acc: 0.8273
Epoch 37/40
996/996 [==============================] - 67s 67ms/step - loss: 0.0011 - model_acc: 0.9831 - val_loss: 0.0126 - val_mode
l_acc: 0.8464
Epoch 38/40
996/996 [==============================] - 68s 68ms/step - loss: 5.8611e-04 - model_acc: 0.9930 - val_loss: 0.0126 - val_
model_acc: 0.8475
Epoch 39/40
996/996 [==============================] - 69s 69ms/step - loss: 5.3376e-04 - model_acc: 0.9944 - val_loss: 0.0124 - val_
model_acc: 0.8475
Epoch 40/40
996/996 [==============================] - 68s 68ms/step - loss: 6.7612e-04 - model_acc: 0.9906 - val_loss: 0.0126 - val_
model_acc: 0.8456
```

Out[9]:

```
<tensorflow.python.keras.callbacks.History at 0x2be1cbf97f0>
```

In [ ]: