```
In [1]:   # data augmentation: mirror and rotate +-25 degree (use read_dataset3, dataset3)
          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 = 130967 # 2~130968
          training_samples = 28317*4  # 2~113269 (Training)
          validation_samples = 3541*4 # 113270~127433 (PublicTest)
          test_samples = 3535         # 127434~130968 (PrivateTest)

          image_path = "./dataset3/images.npy"
          emotion_multi_path = "./dataset3/emotions_multi.npy"
          emotion_single_path = "./dataset3/emotions_single.npy"
```

```
In [2]:   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)
```

```
(130967, 48, 48, 1)
(130967, 8)
(130967, 8)
```

```
In [3]:   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 [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)
# 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: (127432, 48, 48, 1)
training_emotions shape: (127432, 8)
test_images shape: (3535, 48, 48, 1)
test_emotions shape: (3535, 8)
```

In [5]:
```
from tensorflow.python.keras.applications import vgg16, resnet_v2
from tensorflow.python.keras import optimizers
from tensorflow.python.keras.optimizer_v2 import adam
```

In [6]:
```
base_model = vgg16.VGG16(include_top=False,
                         weights="imagenet",
                         input_shape=(48,48,3))
```

```python
base_model.trainable=True
model = Sequential([
    base_model,
    layers.GlobalAveragePooling2D(),
    layers.Dense(4096, activation='relu'),
    layers.Dense(4096, activation='relu'),
    layers.Dense(emotions_count, activation='softmax'),
])

model.compile(optimizer=adam.Adam(learning_rate=1e-4),
              loss=losses.MeanSquaredError(),
              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))
```

```
C:\Users\Darkl\anaconda3\lib\site-packages\tensorflow\python\data\ops\dataset_ops.py:3703: UserWarning: Even though the `tf.c
onfig.experimental_run_functions_eagerly` option is set, this option does not apply to tf.data functions. To force eager exec
ution of tf.data functions, please use `tf.data.experimental.enable.debug_mode()`.
  warnings.warn(
Epoch 1/40
3983/3983 [==============================] - 271s 67ms/step - loss: 0.0242 - model_acc: 0.7300 - val_loss: 0.0161 - val_model
_acc: 0.8005
Epoch 2/40
3983/3983 [==============================] - 270s 68ms/step - loss: 0.0148 - model_acc: 0.8176 - val_loss: 0.0145 - val_model
_acc: 0.8216
Epoch 3/40
3983/3983 [==============================] - 253s 64ms/step - loss: 0.0115 - model_acc: 0.8535 - val_loss: 0.0128 - val_model
_acc: 0.8323
Epoch 4/40
3983/3983 [==============================] - 256s 64ms/step - loss: 0.0093 - model_acc: 0.8800 - val_loss: 0.0127 - val_model
_acc: 0.8391
Epoch 5/40
3983/3983 [==============================] - 260s 65ms/step - loss: 0.0076 - model_acc: 0.9007 - val_loss: 0.0125 - val_model
_acc: 0.8489
Epoch 6/40
3983/3983 [==============================] - 259s 65ms/step - loss: 0.0065 - model_acc: 0.9159 - val_loss: 0.0122 - val_model
_acc: 0.8428
Epoch 7/40
3983/3983 [==============================] - 252s 63ms/step - loss: 0.0055 - model_acc: 0.9293 - val_loss: 0.0122 - val_model
_acc: 0.8520
```

```
Epoch 8/40
3983/3983 [==============================] - 255s 64ms/step - loss: 0.0048 - model_acc: 0.9370 - val_loss: 0.0123 - val_model
_acc: 0.8440
Epoch 9/40
3983/3983 [==============================] - 256s 64ms/step - loss: 0.0042 - model_acc: 0.9448 - val_loss: 0.0124 - val_model
_acc: 0.8448
Epoch 10/40
3983/3983 [==============================] - 250s 63ms/step - loss: 0.0038 - model_acc: 0.9508 - val_loss: 0.0113 - val_model
_acc: 0.8546
Epoch 11/40
3983/3983 [==============================] - 250s 63ms/step - loss: 0.0033 - model_acc: 0.9563 - val_loss: 0.0116 - val_model
_acc: 0.8506
Epoch 12/40
3983/3983 [==============================] - 256s 64ms/step - loss: 0.0031 - model_acc: 0.9583 - val_loss: 0.0116 - val_model
_acc: 0.8479
Epoch 13/40
3983/3983 [==============================] - 253s 63ms/step - loss: 0.0027 - model_acc: 0.9639 - val_loss: 0.0117 - val_model
_acc: 0.8518
Epoch 14/40
3983/3983 [==============================] - 253s 63ms/step - loss: 0.0026 - model_acc: 0.9654 - val_loss: 0.0114 - val_model
_acc: 0.8538
Epoch 15/40
3983/3983 [==============================] - 254s 64ms/step - loss: 0.0024 - model_acc: 0.9673 - val_loss: 0.0115 - val_model
_acc: 0.8510
Epoch 16/40
3983/3983 [==============================] - 256s 64ms/step - loss: 0.0022 - model_acc: 0.9706 - val_loss: 0.0117 - val_model
_acc: 0.8524
Epoch 17/40
3983/3983 [==============================] - 251s 63ms/step - loss: 0.0020 - model_acc: 0.9725 - val_loss: 0.0114 - val_model
_acc: 0.8566
Epoch 18/40
3983/3983 [==============================] - 253s 63ms/step - loss: 0.0019 - model_acc: 0.9727 - val_loss: 0.0116 - val_model
_acc: 0.8566
Epoch 19/40
3983/3983 [==============================] - 255s 64ms/step - loss: 0.0017 - model_acc: 0.9767 - val_loss: 0.0117 - val_model
_acc: 0.8487
Epoch 20/40
3983/3983 [==============================] - 255s 64ms/step - loss: 0.0016 - model_acc: 0.9777 - val_loss: 0.0116 - val_model
_acc: 0.8557
Epoch 21/40
3983/3983 [==============================] - 250s 63ms/step - loss: 0.0018 - model_acc: 0.9742 - val_loss: 0.0115 - val_model
_acc: 0.8557
Epoch 22/40
3983/3983 [==============================] - 252s 63ms/step - loss: 0.0013 - model_acc: 0.9826 - val_loss: 0.0115 - val_model
```

```
_acc: 0.8560
Epoch 23/40
3983/3983 [==============================] - 256s 64ms/step - loss: 0.0013 - model_acc: 0.9820 - val_loss: 0.0116 - val_model
_acc: 0.8549
Epoch 24/40
3983/3983 [==============================] - 251s 63ms/step - loss: 0.0015 - model_acc: 0.9779 - val_loss: 0.0117 - val_model
_acc: 0.8532
Epoch 25/40
3983/3983 [==============================] - 252s 63ms/step - loss: 0.0011 - model_acc: 0.9855 - val_loss: 0.0113 - val_model
_acc: 0.8543
Epoch 26/40
3983/3983 [==============================] - 255s 64ms/step - loss: 0.0014 - model_acc: 0.9801 - val_loss: 0.0126 - val_model
_acc: 0.8473
Epoch 27/40
3983/3983 [==============================] - 256s 64ms/step - loss: 0.0011 - model_acc: 0.9855 - val_loss: 0.0117 - val_model
_acc: 0.8549
Epoch 28/40
3983/3983 [==============================] - 250s 63ms/step - loss: 0.0012 - model_acc: 0.9839 - val_loss: 0.0125 - val_model
_acc: 0.8521
Epoch 29/40
3983/3983 [==============================] - 253s 64ms/step - loss: 0.0012 - model_acc: 0.9828 - val_loss: 0.0113 - val_model
_acc: 0.8574
Epoch 30/40
3983/3983 [==============================] - 256s 64ms/step - loss: 8.3470e-04 - model_acc: 0.9899 - val_loss: 0.0119 - val_m
odel_acc: 0.8515
Epoch 31/40
3983/3983 [==============================] - 253s 63ms/step - loss: 9.9303e-04 - model_acc: 0.9865 - val_loss: 0.0117 - val_m
odel_acc: 0.8569
Epoch 32/40
3983/3983 [==============================] - 250s 63ms/step - loss: 0.0010 - model_acc: 0.9858 - val_loss: 0.0119 - val_model
_acc: 0.8529
Epoch 33/40
3983/3983 [==============================] - 255s 64ms/step - loss: 7.7120e-04 - model_acc: 0.9901 - val_loss: 0.0121 - val_m
odel_acc: 0.8521
Epoch 34/40
3983/3983 [==============================] - 256s 64ms/step - loss: 8.8429e-04 - model_acc: 0.9881 - val_loss: 0.0118 - val_m
odel_acc: 0.8602
Epoch 35/40
3983/3983 [==============================] - 250s 63ms/step - loss: 9.1974e-04 - model_acc: 0.9874 - val_loss: 0.0120 - val_m
odel_acc: 0.8479
Epoch 36/40
3983/3983 [==============================] - 253s 63ms/step - loss: 7.8259e-04 - model_acc: 0.9896 - val_loss: 0.0122 - val_m
odel_acc: 0.8475
Epoch 37/40
```

```
3983/3983 [==============================] - 256s 64ms/step - loss: 6.7970e-04 - model_acc: 0.9916 - val_loss: 0.0124 - val_m
odel_acc: 0.8461
Epoch 38/40
3983/3983 [==============================] - 254s 64ms/step - loss: 8.8567e-04 - model_acc: 0.9885 - val_loss: 0.0127 - val_m
odel_acc: 0.8504
Epoch 39/40
3983/3983 [==============================] - 251s 63ms/step - loss: 9.0546e-04 - model_acc: 0.9874 - val_loss: 0.0123 - val_m
odel_acc: 0.8510
Epoch 40/40
3983/3983 [==============================] - 254s 64ms/step - loss: 5.7754e-04 - model_acc: 0.9930 - val_loss: 0.0120 - val_m
odel_acc: 0.8557
```

Out[6]:   `<tensorflow.python.keras.callbacks.History at 0x22c00904610>`

In [7]:
```python
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(4096, activation='relu'),
    layers.Dense(4096, activation='relu'),
    layers.Dense(emotions_count, activation='softmax'),
])

model.compile(optimizer=adam.Adam(learning_rate=1e-4),
              loss=losses.CategoricalCrossentropy(),
              metrics = "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))
```

```
---------------------------------------------------------------------------
ResourceExhaustedError                    Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_10948/164833257.py in <module>
     15               metrics = "acc")
     16
---> 17 model.fit(x=tf.image.grayscale_to_rgb(training_images),
     18           y=training_emotions,
     19           batch_size=32,
```