In [1]:
```
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 [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)
```

```
(35393, 48, 48, 1)
(35393, 8)
(35393, 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: (31858, 48, 48, 3)
training_emotions shape: (31858, 8)
test_images shape: (3535, 48, 48, 3)
test_emotions shape: (3535, 8)
```

In [5]:

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

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=25,
          validation_data=(test_images, test_emotions))
```

```
C:\Users\Darkl\anaconda3\lib\site-packages\tensorflow\python\data\ops\dataset_ops.py:3703: UserWarning: Even though the `tf.confi
g.experimental_run_functions_eagerly` option is set, this option does not apply to tf.data functions. To force eager execution of
tf.data functions, please use `tf.data.experimental.enable.debug_mode()`.
  warnings.warn(
Epoch 1/25
996/996 [==============================] - 72s 70ms/step - loss: 1.1318 - model_acc: 0.6998 - val_loss: 0.9739 - val_model_acc: 0.
7542
Epoch 2/25
996/996 [==============================] - 69s 70ms/step - loss: 0.9351 - model_acc: 0.7955 - val_loss: 0.9322 - val_model_acc: 0.
7852
Epoch 3/25
996/996 [==============================] - 69s 70ms/step - loss: 0.8601 - model_acc: 0.8374 - val_loss: 0.9220 - val_model_acc: 0.
7819
Epoch 4/25
996/996 [==============================] - 70s 70ms/step - loss: 0.8126 - model_acc: 0.8648 - val_loss: 0.8738 - val_model_acc: 0.
8096
Epoch 5/25
996/996 [==============================] - 69s 70ms/step - loss: 0.7694 - model_acc: 0.8907 - val_loss: 0.8603 - val_model_acc: 0.
8227
Epoch 6/25
996/996 [==============================] - 70s 70ms/step - loss: 0.7411 - model_acc: 0.9074 - val_loss: 0.8667 - val_model_acc: 0.
8134
Epoch 7/25
996/996 [==============================] - 69s 70ms/step - loss: 0.7169 - model_acc: 0.9202 - val_loss: 0.8403 - val_model_acc: 0.
8295
Epoch 8/25
996/996 [==============================] - 69s 69ms/step - loss: 0.6946 - model_acc: 0.9286 - val_loss: 0.8507 - val_model_acc: 0.
```

```
8323
Epoch 9/25
996/996 [==============================] - 70s 70ms/step - loss: 0.6751 - model_acc: 0.9324 - val_loss: 0.8586 - val_model_acc: 0.
8306
Epoch 10/25
996/996 [==============================] - 69s 70ms/step - loss: 0.6590 - model_acc: 0.9362 - val_loss: 0.8734 - val_model_acc: 0.
8261
Epoch 11/25
996/996 [==============================] - 71s 71ms/step - loss: 0.6415 - model_acc: 0.9418 - val_loss: 0.8744 - val_model_acc: 0.
8357
Epoch 12/25
996/996 [==============================] - 68s 68ms/step - loss: 0.6240 - model_acc: 0.9468 - val_loss: 0.9043 - val_model_acc: 0.
8261
Epoch 13/25
996/996 [==============================] - 68s 68ms/step - loss: 0.6105 - model_acc: 0.9496 - val_loss: 0.9231 - val_model_acc: 0.
8287
Epoch 14/25
996/996 [==============================] - 68s 68ms/step - loss: 0.6012 - model_acc: 0.9503 - val_loss: 0.9119 - val_model_acc: 0.
8323
Epoch 15/25
996/996 [==============================] - 68s 69ms/step - loss: 0.5926 - model_acc: 0.9536 - val_loss: 0.9539 - val_model_acc: 0.
8329
Epoch 16/25
996/996 [==============================] - 68s 68ms/step - loss: 0.5868 - model_acc: 0.9541 - val_loss: 0.9647 - val_model_acc: 0.
8340
Epoch 17/25
996/996 [==============================] - 68s 68ms/step - loss: 0.5822 - model_acc: 0.9561 - val_loss: 0.9713 - val_model_acc: 0.
8315
Epoch 18/25
996/996 [==============================] - 68s 68ms/step - loss: 0.5785 - model_acc: 0.9582 - val_loss: 0.9813 - val_model_acc: 0.
8345
Epoch 19/25
996/996 [==============================] - 68s 68ms/step - loss: 0.5740 - model_acc: 0.9594 - val_loss: 1.0144 - val_model_acc: 0.
8270
Epoch 20/25
996/996 [==============================] - 68s 68ms/step - loss: 0.5732 - model_acc: 0.9589 - val_loss: 0.9886 - val_model_acc: 0.
8304
Epoch 21/25
996/996 [==============================] - 68s 68ms/step - loss: 0.5718 - model_acc: 0.9583 - val_loss: 0.9980 - val_model_acc: 0.
8340
Epoch 22/25
996/996 [==============================] - 68s 68ms/step - loss: 0.5695 - model_acc: 0.9602 - val_loss: 1.0179 - val_model_acc: 0.
8357
Epoch 23/25
```

```
996/996 [==============================] - 68s 68ms/step - loss: 0.5667 - model_acc: 0.9636 - val_loss: 1.0281 - val_model_acc: 0.
8337
Epoch 24/25
996/996 [==============================] - 68s 68ms/step - loss: 0.5652 - model_acc: 0.9645 - val_loss: 1.0620 - val_model_acc: 0.
8366
Epoch 25/25
996/996 [==============================] - 68s 68ms/step - loss: 0.5643 - model_acc: 0.9634 - val_loss: 1.0459 - val_model_acc: 0.
8340
```

Out[5]: <tensorflow.python.keras.callbacks.History at 0x1eb808d0250>

In [ ]: