In [1]:

```
import numpy as np
import matplotlib.pyplot as plt


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
from tensorflow.python.keras.layers import Dense, GlobalAveragePooling2D, MaxPool2D, Input, Conv2D, Flatten
from tensorflow.python.keras.models import Model
from tensorflow.python.keras import layers, Sequential,losses, metrics
from tensorflow.python.keras import optimizers, callbacks, models
from tensorflow.python.keras.optimizer_v2 import adam
```

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, densenet, efficientnet
from tensorflow.python.keras.layers import Dense, GlobalAveragePooling2D, MaxPool2D, Input, Conv2D, Flatten, Concatenate, Dropout
from tensorflow.python.keras.models import Model
from tensorflow.python.keras import layers, Sequential

# VGG13 combined .5dropout
input_layer = Input(shape=(48,48,3))
print(input_layer.shape)
feat1 = GlobalAveragePooling2D()(input_layer)
print("feature1", feat1.shape)
x = Conv2D (filters=64, kernel_size=3, padding='same', activation='relu')(input_layer)
x = Conv2D (filters=64, kernel_size=3, padding='same', activation='relu')(x)
x = MaxPool2D(pool_size=2, strides=2, padding='same')(x)
print(x.shape)
feat2 = GlobalAveragePooling2D()(x)
print("feat2", feat2.shape)
x = Conv2D (filters=128, kernel_size=3, padding='same', activation='relu')(x)
x = Conv2D (filters=128, kernel_size=3, padding='same', activation='relu')(x)
x = MaxPool2D(pool_size=2, strides=2, padding='same')(x)
print(x.shape)
feat3 = GlobalAveragePooling2D()(x)
print("feature3", feat3.shape)
x = Conv2D (filters=256, kernel_size=3, padding='same', activation='relu')(x)
x = Conv2D (filters=256, kernel_size=3, padding='same', activation='relu')(x)
#x = Conv2D (filters=256, kernel_size=3, padding='same', activation='relu')(x)
x = MaxPool2D(pool_size=2, strides=2, padding='same')(x)
print(x.shape)
feat4 = GlobalAveragePooling2D()(x)
print("feature4", feat4.shape)
x = Conv2D (filters=512, kernel_size=3, padding='same', activation='relu')(x)
x = Conv2D (filters=512, kernel_size=3, padding='same', activation='relu')(x)
#x = Conv2D (filters=512, kernel_size=3, padding='same', activation='relu')(x)
```

```
x = MaxPool2D(pool_size=2, strides=2, padding='same')(x)
print(x.shape)
feat5 = GlobalAveragePooling2D()(x)
print("feature5", feat5.shape)
x = Conv2D (filters=512, kernel_size=3, padding='same', activation='relu')(x)
x = Conv2D (filters=512, kernel_size=3, padding='same', activation='relu')(x)
#x = Conv2D (filters=512, kernel_size=3, padding='same', activation='relu')(x)
#x = MaxPool2D(pool_size=2, strides=2, padding='same')(x)
#print(x.shape)
x = GlobalAveragePooling2D()(x)
print(x.shape)
feat6 = x
print("feature6", feat6.shape)

x = tf.concat([feat1, feat2, feat3, feat4, feat5, feat6], -1)
print("combined feature", x.shape)
x = Dense(units=4096, activation='relu')(x)
x = Dropout(0.5)(x)
x = Dense(units=4096, activation='relu')(x)
x = Dropout(0.5)(x)
output_layer = Dense(units=8, activation='softmax')(x)
model = Model(inputs=input_layer, outputs=output_layer)
model.summary()

model.compile(optimizer=adam.Adam(learning_rate=2e-4),
              loss=mse,
              metrics = [model_acc])

model.fit(x=tf.image.grayscale_to_rgb(training_images),
          y=training_emotions,
          batch_size=32,
          epochs=30,
          validation_data=(tf.image.grayscale_to_rgb(test_images), test_emotions))

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=30,
          validation_data=(tf.image.grayscale_to_rgb(test_images), test_emotions))
```

```
(None, 48, 48, 3)
feature1 (None, 3)
(None, 24, 24, 64)
feature2 (None, 64)
(None, 12, 12, 128)
feature3 (None, 128)
(None, 6, 6, 256)
feature4 (None, 256)
(None, 3, 3, 512)
feature5 (None, 512)
(None, 512)
feature6 (None, 512)
combined feature (None, 1475)
Model: "model"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 48, 48, 3)]  0

__________________________________________________________________________________________________
conv2d (Conv2D)                 (None, 48, 48, 64)   1792        input_1[0][0]

__________________________________________________________________________________________________
conv2d_1 (Conv2D)               (None, 48, 48, 64)   36928       conv2d[0][0]

__________________________________________________________________________________________________
max_pooling2d (MaxPooling2D)    (None, 24, 24, 64)   0           conv2d_1[0][0]

__________________________________________________________________________________________________
conv2d_2 (Conv2D)               (None, 24, 24, 128)  73856       max_pooling2d[0][0]

__________________________________________________________________________________________________
conv2d_3 (Conv2D)               (None, 24, 24, 128)  147584      conv2d_2[0][0]

__________________________________________________________________________________________________
max_pooling2d_1 (MaxPooling2D)  (None, 12, 12, 128)  0           conv2d_3[0][0]

__________________________________________________________________________________________________
conv2d_4 (Conv2D)               (None, 12, 12, 256)  295168      max_pooling2d_1[0][0]

__________________________________________________________________________________________________
conv2d_5 (Conv2D)               (None, 12, 12, 256)  590080      conv2d_4[0][0]

__________________________________________________________________________________________________
max_pooling2d_2 (MaxPooling2D)  (None, 6, 6, 256)    0           conv2d_5[0][0]

__________________________________________________________________________________________________
conv2d_6 (Conv2D)               (None, 6, 6, 512)    1180160     max_pooling2d_2[0][0]

__________________________________________________________________________________________________
conv2d_7 (Conv2D)               (None, 6, 6, 512)    2359808     conv2d_6[0][0]

__________________________________________________________________________________________________
max_pooling2d_3 (MaxPooling2D)  (None, 3, 3, 512)    0           conv2d_7[0][0]

__________________________________________________________________________________________________
conv2d_8 (Conv2D)               (None, 3, 3, 512)    2359808     max_pooling2d_3[0][0]
```

```
__________________________________________________________________________________________________
conv2d_9 (Conv2D)               (None, 3, 3, 512)    2359808     conv2d_8[0][0]
__________________________________________________________________________________________________
global_average_pooling2d (Globa (None, 3)            0           input_1[0][0]
__________________________________________________________________________________________________
global_average_pooling2d_1 (Glo (None, 64)           0           max_pooling2d[0][0]
__________________________________________________________________________________________________
global_average_pooling2d_2 (Glo (None, 128)          0           max_pooling2d_1[0][0]
__________________________________________________________________________________________________
global_average_pooling2d_3 (Glo (None, 256)          0           max_pooling2d_2[0][0]
__________________________________________________________________________________________________
global_average_pooling2d_4 (Glo (None, 512)          0           max_pooling2d_3[0][0]
__________________________________________________________________________________________________
global_average_pooling2d_5 (Glo (None, 512)          0           conv2d_9[0][0]
__________________________________________________________________________________________________
tf.concat (TFOpLambda)          (None, 1475)         0           global_average_pooling2d[0][0]
                                                                 global_average_pooling2d_1[0][0]
                                                                 global_average_pooling2d_2[0][0]
                                                                 global_average_pooling2d_3[0][0]
                                                                 global_average_pooling2d_4[0][0]
                                                                 global_average_pooling2d_5[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 4096)         6045696     tf.concat[0][0]
__________________________________________________________________________________________________
dropout (Dropout)               (None, 4096)         0           dense[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 4096)         16781312    dropout[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 4096)         0           dense_1[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)                 (None, 8)            32776       dropout_1[0][0]
==================================================================================================
Total params: 32,264,776
Trainable params: 32,264,776
Non-trainable params: 0
__________________________________________________________________________________________________
```

```
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 1/30
996/996 [==============================] - 72s 67ms/step - loss: 0.0572 - model_acc: 0.3794 - val_loss: 0.0448 - val_model_acc: 0.
5470
```

```
Epoch 2/30
996/996 [==============================] - 73s 73ms/step - loss: 0.0333 - model_acc: 0.6453 - val_loss: 0.0298 - val_model_acc: 0.
6810
Epoch 3/30
996/996 [==============================] - 73s 73ms/step - loss: 0.0241 - model_acc: 0.7297 - val_loss: 0.0228 - val_model_acc: 0.
7393
Epoch 4/30
996/996 [==============================] - 69s 70ms/step - loss: 0.0192 - model_acc: 0.7743 - val_loss: 0.0196 - val_model_acc: 0.
7585
Epoch 5/30
996/996 [==============================] - 71s 71ms/step - loss: 0.0157 - model_acc: 0.8113 - val_loss: 0.0185 - val_model_acc: 0.
7803
Epoch 6/30
996/996 [==============================] - 72s 72ms/step - loss: 0.0131 - model_acc: 0.8397 - val_loss: 0.0176 - val_model_acc: 0.
7887
Epoch 7/30
996/996 [==============================] - 72s 72ms/step - loss: 0.0110 - model_acc: 0.8638 - val_loss: 0.0171 - val_model_acc: 0.
7922
Epoch 8/30
996/996 [==============================] - 71s 71ms/step - loss: 0.0092 - model_acc: 0.8855 - val_loss: 0.0162 - val_model_acc: 0.
7953
Epoch 9/30
996/996 [==============================] - 73s 73ms/step - loss: 0.0080 - model_acc: 0.8970 - val_loss: 0.0166 - val_model_acc: 0.
8056
Epoch 10/30
996/996 [==============================] - 72s 72ms/step - loss: 0.0068 - model_acc: 0.9146 - val_loss: 0.0164 - val_model_acc: 0.
7984
Epoch 11/30
996/996 [==============================] - 71s 71ms/step - loss: 0.0061 - model_acc: 0.9238 - val_loss: 0.0155 - val_model_acc: 0.
8095
Epoch 12/30
996/996 [==============================] - 68s 68ms/step - loss: 0.0054 - model_acc: 0.9313 - val_loss: 0.0152 - val_model_acc: 0.
8131
Epoch 13/30
996/996 [==============================] - 71s 72ms/step - loss: 0.0048 - model_acc: 0.9404 - val_loss: 0.0155 - val_model_acc: 0.
8063
Epoch 14/30
996/996 [==============================] - 71s 71ms/step - loss: 0.0045 - model_acc: 0.9405 - val_loss: 0.0154 - val_model_acc: 0.
8157
Epoch 15/30
996/996 [==============================] - 71s 71ms/step - loss: 0.0042 - model_acc: 0.9463 - val_loss: 0.0154 - val_model_acc: 0.
8165
Epoch 16/30
996/996 [==============================] - 72s 72ms/step - loss: 0.0039 - model_acc: 0.9492 - val_loss: 0.0154 - val_model_acc: 0.
```

```
8111
Epoch 17/30
996/996 [==============================] - 74s 74ms/step - loss: 0.0036 - model_acc: 0.9518 - val_loss: 0.0150 - val_model_acc: 0.
8151
Epoch 18/30
996/996 [==============================] - 71s 71ms/step - loss: 0.0034 - model_acc: 0.9558 - val_loss: 0.0148 - val_model_acc: 0.
8171
Epoch 19/30
996/996 [==============================] - 69s 69ms/step - loss: 0.0032 - model_acc: 0.9581 - val_loss: 0.0142 - val_model_acc: 0.
8222
Epoch 20/30
996/996 [==============================] - 69s 69ms/step - loss: 0.0031 - model_acc: 0.9577 - val_loss: 0.0148 - val_model_acc: 0.
8162
Epoch 21/30
996/996 [==============================] - 69s 69ms/step - loss: 0.0028 - model_acc: 0.9627 - val_loss: 0.0148 - val_model_acc: 0.
8140
Epoch 22/30
996/996 [==============================] - 84s 85ms/step - loss: 0.0028 - model_acc: 0.9632 - val_loss: 0.0147 - val_model_acc: 0.
8185
Epoch 23/30
996/996 [==============================] - 587s 590ms/step - loss: 0.0025 - model_acc: 0.9672 - val_loss: 0.0142 - val_model_acc:
0.8218
Epoch 24/30
996/996 [==============================] - 482s 483ms/step - loss: 0.0026 - model_acc: 0.9657 - val_loss: 0.0144 - val_model_acc:
0.8162
Epoch 25/30
996/996 [==============================] - 70s 71ms/step - loss: 0.0024 - model_acc: 0.9664 - val_loss: 0.0144 - val_model_acc: 0.
8163
Epoch 26/30
996/996 [==============================] - 68s 68ms/step - loss: 0.0022 - model_acc: 0.9710 - val_loss: 0.0145 - val_model_acc: 0.
8241
Epoch 27/30
996/996 [==============================] - 69s 70ms/step - loss: 0.0021 - model_acc: 0.9717 - val_loss: 0.0145 - val_model_acc: 0.
8236
Epoch 28/30
996/996 [==============================] - 66s 66ms/step - loss: 0.0021 - model_acc: 0.9694 - val_loss: 0.0146 - val_model_acc: 0.
8241
Epoch 29/30
996/996 [==============================] - 67s 67ms/step - loss: 0.0020 - model_acc: 0.9720 - val_loss: 0.0141 - val_model_acc: 0.
8219
Epoch 30/30
996/996 [==============================] - 66s 66ms/step - loss: 0.0019 - model_acc: 0.9752 - val_loss: 0.0143 - val_model_acc: 0.
8239
Epoch 1/30
```

```
996/996 [==============================] - 69s 69ms/step - loss: 0.0015 - model_acc: 0.9796 - val_loss: 0.0137 - val_model_acc: 0.
8269
Epoch 2/30
996/996 [==============================] - 67s 67ms/step - loss: 0.0012 - model_acc: 0.9880 - val_loss: 0.0138 - val_model_acc: 0.
8275
Epoch 3/30
996/996 [==============================] - 68s 68ms/step - loss: 0.0011 - model_acc: 0.9901 - val_loss: 0.0136 - val_model_acc: 0.
8272
Epoch 4/30
996/996 [==============================] - 66s 66ms/step - loss: 0.0010 - model_acc: 0.9904 - val_loss: 0.0138 - val_model_acc: 0.
8289
Epoch 5/30
996/996 [==============================] - 66s 66ms/step - loss: 0.0010 - model_acc: 0.9891 - val_loss: 0.0137 - val_model_acc: 0.
8272
Epoch 6/30
996/996 [==============================] - 68s 68ms/step - loss: 9.7654e-04 - model_acc: 0.9890 - val_loss: 0.0137 - val_model_ac
c: 0.8249
Epoch 7/30
996/996 [==============================] - 66s 66ms/step - loss: 9.3429e-04 - model_acc: 0.9892 - val_loss: 0.0137 - val_model_ac
c: 0.8292
Epoch 8/30
996/996 [==============================] - 67s 67ms/step - loss: 8.9927e-04 - model_acc: 0.9905 - val_loss: 0.0139 - val_model_ac
c: 0.8311
Epoch 9/30
996/996 [==============================] - 66s 67ms/step - loss: 8.7202e-04 - model_acc: 0.9897 - val_loss: 0.0137 - val_model_ac
c: 0.8272
Epoch 10/30
996/996 [==============================] - 67s 67ms/step - loss: 8.3579e-04 - model_acc: 0.9907 - val_loss: 0.0139 - val_model_ac
c: 0.8249
Epoch 11/30
996/996 [==============================] - 66s 66ms/step - loss: 7.9788e-04 - model_acc: 0.9906 - val_loss: 0.0137 - val_model_ac
c: 0.8275
Epoch 12/30
996/996 [==============================] - 66s 66ms/step - loss: 7.7263e-04 - model_acc: 0.9907 - val_loss: 0.0137 - val_model_ac
c: 0.8269
Epoch 13/30
996/996 [==============================] - 66s 66ms/step - loss: 7.5265e-04 - model_acc: 0.9919 - val_loss: 0.0139 - val_model_ac
c: 0.8247
Epoch 14/30
996/996 [==============================] - 66s 66ms/step - loss: 7.3008e-04 - model_acc: 0.9910 - val_loss: 0.0138 - val_model_ac
c: 0.8297
Epoch 15/30
996/996 [==============================] - 66s 66ms/step - loss: 7.1540e-04 - model_acc: 0.9913 - val_loss: 0.0137 - val_model_ac
c: 0.8241
```

```
Epoch 16/30
996/996 [==============================] - 66s 66ms/step - loss: 6.9342e-04 - model_acc: 0.9926 - val_loss: 0.0137 - val_model_ac
c: 0.8269
Epoch 17/30
996/996 [==============================] - 66s 67ms/step - loss: 6.7100e-04 - model_acc: 0.9907 - val_loss: 0.0136 - val_model_ac
c: 0.8252
Epoch 18/30
996/996 [==============================] - 66s 67ms/step - loss: 6.5390e-04 - model_acc: 0.9921 - val_loss: 0.0136 - val_model_ac
c: 0.8261
Epoch 19/30
996/996 [==============================] - 66s 67ms/step - loss: 6.3576e-04 - model_acc: 0.9916 - val_loss: 0.0137 - val_model_ac
c: 0.8281
Epoch 20/30
996/996 [==============================] - 66s 66ms/step - loss: 6.2902e-04 - model_acc: 0.9919 - val_loss: 0.0137 - val_model_ac
c: 0.8300
Epoch 21/30
996/996 [==============================] - 66s 66ms/step - loss: 6.0865e-04 - model_acc: 0.9917 - val_loss: 0.0136 - val_model_ac
c: 0.8266
Epoch 22/30
996/996 [==============================] - 66s 66ms/step - loss: 5.9049e-04 - model_acc: 0.9929 - val_loss: 0.0137 - val_model_ac
c: 0.8266
Epoch 23/30
996/996 [==============================] - 67s 67ms/step - loss: 5.8362e-04 - model_acc: 0.9919 - val_loss: 0.0136 - val_model_ac
c: 0.8292
Epoch 24/30
996/996 [==============================] - 66s 66ms/step - loss: 5.6745e-04 - model_acc: 0.9930 - val_loss: 0.0136 - val_model_ac
c: 0.8264
Epoch 25/30
996/996 [==============================] - 66s 66ms/step - loss: 5.6003e-04 - model_acc: 0.9934 - val_loss: 0.0138 - val_model_ac
c: 0.8283
Epoch 26/30
996/996 [==============================] - 66s 66ms/step - loss: 5.4616e-04 - model_acc: 0.9931 - val_loss: 0.0137 - val_model_ac
c: 0.8261
Epoch 27/30
996/996 [==============================] - 65s 66ms/step - loss: 5.3248e-04 - model_acc: 0.9924 - val_loss: 0.0136 - val_model_ac
c: 0.8275
Epoch 28/30
996/996 [==============================] - 66s 66ms/step - loss: 5.2359e-04 - model_acc: 0.9931 - val_loss: 0.0137 - val_model_ac
c: 0.8292
Epoch 29/30
996/996 [==============================] - 65s 65ms/step - loss: 5.1446e-04 - model_acc: 0.9929 - val_loss: 0.0137 - val_model_ac
c: 0.8286
Epoch 30/30
```