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]:

```
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 [3]:

```
#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 [4]:

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

In [5]:

```
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 [6]:

```
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 [7]:

```
from tensorflow.python.keras.applications import vgg16, resnet_v2, densenet, efficientnet, inception_resnet_v2, inception_v3, nasn
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

base_model = densenet.DenseNet121(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.Dropout(0.5),
    layers.Dense(4096, activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(emotions_count, activation='softmax'),
])

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))
```

```
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 [==============================] - 289s 288ms/step - loss: 0.0392 - model_acc: 0.6082 - val_loss: 0.0259 - val_model_acc:
0.7219
Epoch 2/30
996/996 [==============================] - 281s 282ms/step - loss: 0.0223 - model_acc: 0.7529 - val_loss: 0.0216 - val_model_acc:
0.7571
Epoch 3/30
996/996 [==============================] - 280s 281ms/step - loss: 0.0175 - model_acc: 0.7977 - val_loss: 0.0186 - val_model_acc:
0.7794
Epoch 4/30
```

```
996/996 [==============================] - 280s 281ms/step - loss: 0.0148 - model_acc: 0.8258 - val_loss: 0.0192 - val_model_acc:
0.7706
Epoch 5/30
996/996 [==============================] - 280s 281ms/step - loss: 0.0128 - model_acc: 0.8447 - val_loss: 0.0184 - val_model_acc:
0.7867
Epoch 6/30
996/996 [==============================] - 284s 285ms/step - loss: 0.0112 - model_acc: 0.8620 - val_loss: 0.0176 - val_model_acc:
0.7917
Epoch 7/30
996/996 [==============================] - 280s 281ms/step - loss: 0.0102 - model_acc: 0.8743 - val_loss: 0.0188 - val_model_acc:
0.7830
Epoch 8/30
996/996 [==============================] - 281s 283ms/step - loss: 0.0089 - model_acc: 0.8901 - val_loss: 0.0164 - val_model_acc:
0.8126
Epoch 9/30
996/996 [==============================] - 294s 296ms/step - loss: 0.0083 - model_acc: 0.8989 - val_loss: 0.0163 - val_model_acc:
0.8050
Epoch 10/30
996/996 [==============================] - 296s 297ms/step - loss: 0.0076 - model_acc: 0.9065 - val_loss: 0.0169 - val_model_acc:
0.8052
Epoch 11/30
996/996 [==============================] - 296s 297ms/step - loss: 0.0074 - model_acc: 0.9080 - val_loss: 0.0184 - val_model_acc:
0.7948
Epoch 12/30
996/996 [==============================] - 293s 294ms/step - loss: 0.0073 - model_acc: 0.9099 - val_loss: 0.0158 - val_model_acc:
0.8171
Epoch 13/30
996/996 [==============================] - 286s 287ms/step - loss: 0.0059 - model_acc: 0.9275 - val_loss: 0.0152 - val_model_acc:
0.8207
Epoch 14/30
996/996 [==============================] - 286s 287ms/step - loss: 0.0057 - model_acc: 0.9279 - val_loss: 0.0170 - val_model_acc:
0.7987
Epoch 15/30
996/996 [==============================] - 297s 298ms/step - loss: 0.0056 - model_acc: 0.9318 - val_loss: 0.0149 - val_model_acc:
0.8250
Epoch 16/30
996/996 [==============================] - 298s 299ms/step - loss: 0.0054 - model_acc: 0.9326 - val_loss: 0.0162 - val_model_acc:
0.8098
Epoch 17/30
996/996 [==============================] - 297s 299ms/step - loss: 0.0054 - model_acc: 0.9307 - val_loss: 0.0155 - val_model_acc:
0.8188
Epoch 18/30
996/996 [==============================] - 295s 296ms/step - loss: 0.0046 - model_acc: 0.9431 - val_loss: 0.0151 - val_model_acc:
0.8236
```

```
Epoch 19/30
996/996 [==============================] - 295s 296ms/step - loss: 0.0047 - model_acc: 0.9367 - val_loss: 0.0179 - val_model_acc:
0.7963
Epoch 20/30
996/996 [==============================] - 295s 296ms/step - loss: 0.0045 - model_acc: 0.9433 - val_loss: 0.0155 - val_model_acc:
0.8160
Epoch 21/30
996/996 [==============================] - 295s 296ms/step - loss: 0.0039 - model_acc: 0.9493 - val_loss: 0.0150 - val_model_acc:
0.8180
Epoch 22/30
996/996 [==============================] - 294s 295ms/step - loss: 0.0038 - model_acc: 0.9495 - val_loss: 0.0145 - val_model_acc:
0.8239
Epoch 23/30
996/996 [==============================] - 294s 295ms/step - loss: 0.0039 - model_acc: 0.9481 - val_loss: 0.0162 - val_model_acc:
0.8084
Epoch 24/30
996/996 [==============================] - 294s 295ms/step - loss: 0.0037 - model_acc: 0.9509 - val_loss: 0.0148 - val_model_acc:
0.8249
Epoch 25/30
996/996 [==============================] - 285s 287ms/step - loss: 0.0037 - model_acc: 0.9515 - val_loss: 0.0148 - val_model_acc:
0.8210
Epoch 26/30
996/996 [==============================] - 286s 287ms/step - loss: 0.0041 - model_acc: 0.9465 - val_loss: 0.0150 - val_model_acc:
0.8245
Epoch 27/30
996/996 [==============================] - 292s 294ms/step - loss: 0.0031 - model_acc: 0.9590 - val_loss: 0.0148 - val_model_acc:
0.8208
Epoch 28/30
996/996 [==============================] - 300s 301ms/step - loss: 0.0028 - model_acc: 0.9635 - val_loss: 0.0145 - val_model_acc:
0.8163
Epoch 29/30
996/996 [==============================] - 298s 300ms/step - loss: 0.0030 - model_acc: 0.9583 - val_loss: 0.0151 - val_model_acc:
0.8222
Epoch 30/30
996/996 [==============================] - 294s 295ms/step - loss: 0.0042 - model_acc: 0.9414 - val_loss: 0.0143 - val_model_acc:
0.8281
Epoch 1/30
996/996 [==============================] - 294s 295ms/step - loss: 0.0026 - model_acc: 0.9672 - val_loss: 0.0141 - val_model_acc:
0.8258
Epoch 2/30
996/996 [==============================] - 293s 294ms/step - loss: 0.0021 - model_acc: 0.9753 - val_loss: 0.0137 - val_model_acc:
0.8321
Epoch 3/30
996/996 [==============================] - 293s 294ms/step - loss: 0.0019 - model_acc: 0.9777 - val_loss: 0.0137 - val_model_acc:
```

```
0.8369
Epoch 4/30
996/996 [==============================] - 293s 294ms/step - loss: 0.0018 - model_acc: 0.9798 - val_loss: 0.0142 - val_model_acc:
0.8241
Epoch 5/30
996/996 [==============================] - 293s 294ms/step - loss: 0.0018 - model_acc: 0.9776 - val_loss: 0.0139 - val_model_acc:
0.8371
Epoch 6/30
996/996 [==============================] - 293s 294ms/step - loss: 0.0016 - model_acc: 0.9797 - val_loss: 0.0137 - val_model_acc:
0.8301
Epoch 7/30
996/996 [==============================] - 289s 290ms/step - loss: 0.0019 - model_acc: 0.9756 - val_loss: 0.0138 - val_model_acc:
0.8357
Epoch 8/30
996/996 [==============================] - 287s 288ms/step - loss: 0.0015 - model_acc: 0.9838 - val_loss: 0.0136 - val_model_acc:
0.8363
Epoch 9/30
996/996 [==============================] - 295s 296ms/step - loss: 0.0014 - model_acc: 0.9850 - val_loss: 0.0141 - val_model_acc:
0.8310
Epoch 10/30
996/996 [==============================] - 300s 301ms/step - loss: 0.0014 - model_acc: 0.9834 - val_loss: 0.0139 - val_model_acc:
0.8326
Epoch 11/30
996/996 [==============================] - 301s 302ms/step - loss: 0.0014 - model_acc: 0.9824 - val_loss: 0.0137 - val_model_acc:
0.8317
Epoch 12/30
996/996 [==============================] - 292s 294ms/step - loss: 0.0014 - model_acc: 0.9832 - val_loss: 0.0148 - val_model_acc:
0.8219
Epoch 13/30
996/996 [==============================] - 293s 294ms/step - loss: 0.0014 - model_acc: 0.9814 - val_loss: 0.0137 - val_model_acc:
0.8354
Epoch 14/30
996/996 [==============================] - 293s 294ms/step - loss: 0.0013 - model_acc: 0.9850 - val_loss: 0.0139 - val_model_acc:
0.8337
Epoch 15/30
996/996 [==============================] - 292s 293ms/step - loss: 0.0012 - model_acc: 0.9861 - val_loss: 0.0139 - val_model_acc:
0.8318
Epoch 16/30
996/996 [==============================] - 293s 294ms/step - loss: 0.0011 - model_acc: 0.9873 - val_loss: 0.0140 - val_model_acc:
0.8315
Epoch 17/30
996/996 [==============================] - 295s 296ms/step - loss: 0.0012 - model_acc: 0.9857 - val_loss: 0.0138 - val_model_acc:
0.8337
Epoch 18/30
```

```
996/996 [==============================] - 294s 295ms/step - loss: 0.0013 - model_acc: 0.9846 - val_loss: 0.0141 - val_model_acc:
0.8313
Epoch 19/30
996/996 [==============================] - 291s 292ms/step - loss: 0.0011 - model_acc: 0.9875 - val_loss: 0.0140 - val_model_acc:
0.8292
Epoch 20/30
996/996 [==============================] - 288s 289ms/step - loss: 0.0011 - model_acc: 0.9855 - val_loss: 0.0138 - val_model_acc:
0.8346
Epoch 21/30
996/996 [==============================] - 289s 290ms/step - loss: 0.0012 - model_acc: 0.9844 - val_loss: 0.0143 - val_model_acc:
0.8295
Epoch 22/30
996/996 [==============================] - 299s 300ms/step - loss: 0.0011 - model_acc: 0.9860 - val_loss: 0.0136 - val_model_acc:
0.8295
Epoch 23/30
996/996 [==============================] - 299s 300ms/step - loss: 0.0012 - model_acc: 0.9840 - val_loss: 0.0142 - val_model_acc:
0.8297
Epoch 24/30
996/996 [==============================] - 294s 295ms/step - loss: 0.0011 - model_acc: 0.9865 - val_loss: 0.0138 - val_model_acc:
0.8346
Epoch 25/30
996/996 [==============================] - 293s 295ms/step - loss: 9.0149e-04 - model_acc: 0.9902 - val_loss: 0.0138 - val_model_a
cc: 0.8348
Epoch 26/30
996/996 [==============================] - 294s 295ms/step - loss: 9.4518e-04 - model_acc: 0.9897 - val_loss: 0.0144 - val_model_a
cc: 0.8306
Epoch 27/30
996/996 [==============================] - 295s 297ms/step - loss: 9.5081e-04 - model_acc: 0.9876 - val_loss: 0.0137 - val_model_a
cc: 0.8348
Epoch 28/30
996/996 [==============================] - 295s 296ms/step - loss: 9.8644e-04 - model_acc: 0.9872 - val_loss: 0.0140 - val_model_a
cc: 0.8306
Epoch 29/30
996/996 [==============================] - 295s 296ms/step - loss: 9.4516e-04 - model_acc: 0.9868 - val_loss: 0.0139 - val_model_a
cc: 0.8348
Epoch 30/30
996/996 [==============================] - 296s 298ms/step - loss: 8.8750e-04 - model_acc: 0.9882 - val_loss: 0.0138 - val_model_a
cc: 0.8326
```

Out[7]:

```
<tensorflow.python.keras.callbacks.History at 0x19983272f70>
```

In [10]:

```
base_model = nasnet.NASNetMobile(include_top=False, weights="imagenet", input_shape=(48,48,3))
base_model.trainable=True
```