In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
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
from tensorflow.python.keras import losses, metrics
from tensorflow.python.keras.optimizer_v2 import adam
from tensorflow.python.keras.applications import vgg16, resnet_v2, densenet, efficientnet
from tensorflow.python.keras.layers import Dense, GlobalAveragePooling2D, MaxPool2D, Input, Conv2D, Flatten, Concatenate, Dro
from tensorflow.python.keras.models import Model
from tensorflow.python.keras import layers, Sequential
```

In [2]:

```
# data augmentation: mirror and rotate +-25 degree (use read_dataset3, dataset3)
# data augmentation test: rotate different degree (pay attention to adjustable filename etc.)

image_height = 48
image_width = 48
emotions_count = 8
emotion_labels = ['neutral', 'happiness', 'surprise', 'sadness',
                  'anger', 'disgust', 'fear', 'contempt']

# !!! change sample size
samples = 130967 # 2~130968
training_samples = 28317 *4 # 2~113269 (Training)
validation_samples = 3541 *4 # 113270~127433 (PublicTest)
test_samples = 3535 # 127434~130968 (PrivateTest)

# !!! change npy folder name
image_path = "./dataset3/images.npy"
emotion_multi_path = "./dataset3/emotions_multi.npy"
emotion_single_path = "./dataset3/emotions_single.npy"
images = np.load(image_path)
emotions_multi = np.load(emotion_multi_path)
emotions_single = np.load(emotion_single_path)

# !!! change s/m dataset
#emotions = emotions_single
```

```
emotions = emotions_multi
print(images.shape)
print(emotions_multi.shape)
print(emotions_single.shape)
```

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

In [3]:

```
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 [4]:

```
images = tf.convert_to_tensor(images)
emotions = tf.convert_to_tensor(emotions)
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]:

```python
def create_model():
    base_model = vgg16.VGG16(include_top=False,
                             weights="imagenet",
                             input_shape=(48,48,3))
    base_model.trainable=True
    input_layer = Input(shape=(48,48,3))
    print(input_layer.shape)
    feat1 = GlobalAveragePooling2D()(input_layer)
    print("feature1", feat1.shape)
    x = base_model.layers[1](input_layer)

    x = base_model.layers[2](x)
    x = base_model.layers[3](x)
    print(x.shape)
    feat2 = GlobalAveragePooling2D()(x)
    print("feature2", feat2.shape)

    x = base_model.layers[4](x)
    x = base_model.layers[5](x)
    x = base_model.layers[6](x)
    print(x.shape)
    feat3 = GlobalAveragePooling2D()(x)
    print("feature3", feat3.shape)

    x = base_model.layers[7](x)
    x = base_model.layers[8](x)
    x = base_model.layers[9](x)
    x = base_model.layers[10](x)
    print(x.shape)
    feat4 = GlobalAveragePooling2D()(x)
    print("feature4", feat4.shape)

    x = base_model.layers[11](x)
    x = base_model.layers[12](x)
    x = base_model.layers[13](x)
    x = base_model.layers[14](x)
    print(x.shape)
    feat5 = GlobalAveragePooling2D()(x)
    print("feature5", feat5.shape)
    x = base_model.layers[15](x)
    x = base_model.layers[16](x)
    x = base_model.layers[17](x)
    x = base_model.layers[18](x)
    print(x.shape)
    feat6 = GlobalAveragePooling2D()(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)

    return model

model = create_model()
model.summary()
```

```
(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, 1, 1, 512)
feature6 (None, 512)
combined feature (None, 1475)
Model: "model"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_2 (InputLayer)            [(None, 48, 48, 3)]  0
__________________________________________________________________________________________________
```

```
block1_conv1 (Conv2D)           (None, 48, 48, 64)   1792        input_2[0][0]

block1_conv2 (Conv2D)           (None, 48, 48, 64)   36928       block1_conv1[1][0]

block1_pool (MaxPooling2D)      (None, 24, 24, 64)   0           block1_conv2[1][0]

block2_conv1 (Conv2D)           (None, 24, 24, 128)  73856       block1_pool[1][0]

block2_conv2 (Conv2D)           (None, 24, 24, 128)  147584      block2_conv1[1][0]

block2_pool (MaxPooling2D)      (None, 12, 12, 128)  0           block2_conv2[1][0]

block3_conv1 (Conv2D)           (None, 12, 12, 256)  295168      block2_pool[1][0]

block3_conv2 (Conv2D)           (None, 12, 12, 256)  590080      block3_conv1[1][0]

block3_conv3 (Conv2D)           (None, 12, 12, 256)  590080      block3_conv2[1][0]

block3_pool (MaxPooling2D)      (None, 6, 6, 256)    0           block3_conv3[1][0]

block4_conv1 (Conv2D)           (None, 6, 6, 512)    1180160     block3_pool[1][0]

block4_conv2 (Conv2D)           (None, 6, 6, 512)    2359808     block4_conv1[1][0]

block4_conv3 (Conv2D)           (None, 6, 6, 512)    2359808     block4_conv2[1][0]

block4_pool (MaxPooling2D)      (None, 3, 3, 512)    0           block4_conv3[1][0]

block5_conv1 (Conv2D)           (None, 3, 3, 512)    2359808     block4_pool[1][0]

block5_conv2 (Conv2D)           (None, 3, 3, 512)    2359808     block5_conv1[1][0]

block5_conv3 (Conv2D)           (None, 3, 3, 512)    2359808     block5_conv2[1][0]

block5_pool (MaxPooling2D)      (None, 1, 1, 512)    0           block5_conv3[1][0]

global_average_pooling2d (Globa (None, 3)            0           input_2[0][0]

global_average_pooling2d_1 (Glo (None, 64)           0           block1_pool[1][0]

global_average_pooling2d_2 (Glo (None, 128)          0           block2_pool[1][0]

global_average_pooling2d_3 (Glo (None, 256)          0           block3_pool[1][0]
global_average_pooling2d_4 (Glo (None, 512)          0           block4_pool[1][0]
__________________________________________________________________________________________________
global_average_pooling2d_5 (Glo (None, 512)          0           block5_pool[1][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]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 4096)         16781312    dense[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)                 (None, 8)            32776       dense_1[0][0]
==================================================================================================
Total params: 37,574,472
Trainable params: 37,574,472
Non-trainable params: 0
__________________________________________________________________________________________________
```

In [6]:

```python
model.compile(optimizer=adam.Adam(learning_rate=2e-5),
              loss=mse,
              metrics = [model_acc])

model.fit(x=tf.image.grayscale_to_rgb(training_images),
          y=training_emotions,
          batch_size=32,
          epochs=60,
          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 1/60
3983/3983 [==============================] - 287s 71ms/step - loss: 0.0357 - model_acc: 0.6472 - val_loss: 0.0211 - val_model
_acc: 0.7590
Epoch 2/60
3983/3983 [==============================] - 285s 72ms/step - loss: 0.0169 - model_acc: 0.7973 - val_loss: 0.0165 - val_model
_acc: 0.7938
Epoch 3/60
```

```
3983/3983 [==============================] - 283s 71ms/step - loss: 0.0130 - model_acc: 0.8372 - val_loss: 0.0146 - val_model
_acc: 0.8109
Epoch 4/60
3983/3983 [==============================] - 287s 72ms/step - loss: 0.0105 - model_acc: 0.8663 - val_loss: 0.0137 - val_model
_acc: 0.8238
Epoch 5/60
3983/3983 [==============================] - 289s 72ms/step - loss: 0.0086 - model_acc: 0.8888 - val_loss: 0.0129 - val_model
_acc: 0.8360
Epoch 6/60
3983/3983 [==============================] - 287s 72ms/step - loss: 0.0072 - model_acc: 0.9075 - val_loss: 0.0130 - val_model
_acc: 0.8371
Epoch 7/60
3983/3983 [==============================] - 287s 72ms/step - loss: 0.0061 - model_acc: 0.9218 - val_loss: 0.0130 - val_model
_acc: 0.8397
Epoch 8/60
3983/3983 [==============================] - 293s 73ms/step - loss: 0.0052 - model_acc: 0.9331 - val_loss: 0.0127 - val_model
_acc: 0.8380
Epoch 9/60
3983/3983 [==============================] - 286s 72ms/step - loss: 0.0046 - model_acc: 0.9413 - val_loss: 0.0128 - val_model
_acc: 0.8381
Epoch 10/60
3983/3983 [==============================] - 289s 73ms/step - loss: 0.0041 - model_acc: 0.9479 - val_loss: 0.0130 - val_model
_acc: 0.8419
Epoch 11/60
3983/3983 [==============================] - 287s 72ms/step - loss: 0.0036 - model_acc: 0.9529 - val_loss: 0.0125 - val_model
_acc: 0.8448
Epoch 12/60
3983/3983 [==============================] - 286s 72ms/step - loss: 0.0033 - model_acc: 0.9576 - val_loss: 0.0126 - val_model
_acc: 0.8380
Epoch 13/60
3983/3983 [==============================] - 291s 73ms/step - loss: 0.0030 - model_acc: 0.9603 - val_loss: 0.0124 - val_model
_acc: 0.8417
Epoch 14/60
3983/3983 [==============================] - 286s 72ms/step - loss: 0.0027 - model_acc: 0.9635 - val_loss: 0.0123 - val_model
_acc: 0.8414
Epoch 15/60
3983/3983 [==============================] - 289s 73ms/step - loss: 0.0025 - model_acc: 0.9664 - val_loss: 0.0122 - val_model
_acc: 0.8473
Epoch 16/60
3983/3983 [==============================] - 283s 71ms/step - loss: 0.0023 - model_acc: 0.9689 - val_loss: 0.0123 - val_model
_acc: 0.8470
Epoch 17/60
3983/3983 [==============================] - 283s 71ms/step - loss: 0.0022 - model_acc: 0.9708 - val_loss: 0.0124 - val_model
_acc: 0.8431
```

```
Epoch 18/60
3983/3983 [==============================] - 278s 70ms/step - loss: 0.0020 - model_acc: 0.9730 - val_loss: 0.0121 - val_model
_acc: 0.8504
Epoch 19/60
3983/3983 [==============================] - 275s 69ms/step - loss: 0.0019 - model_acc: 0.9749 - val_loss: 0.0120 - val_model
_acc: 0.8467
Epoch 20/60
3983/3983 [==============================] - 278s 70ms/step - loss: 0.0017 - model_acc: 0.9763 - val_loss: 0.0122 - val_model
_acc: 0.8434
Epoch 21/60
3983/3983 [==============================] - 281s 71ms/step - loss: 0.0016 - model_acc: 0.9777 - val_loss: 0.0122 - val_model
_acc: 0.8465
Epoch 22/60
3983/3983 [==============================] - 286s 72ms/step - loss: 0.0015 - model_acc: 0.9791 - val_loss: 0.0119 - val_model
_acc: 0.8487
Epoch 23/60
3983/3983 [==============================] - 288s 72ms/step - loss: 0.0015 - model_acc: 0.9801 - val_loss: 0.0120 - val_model
_acc: 0.8501
Epoch 24/60
3983/3983 [==============================] - 283s 71ms/step - loss: 0.0014 - model_acc: 0.9818 - val_loss: 0.0120 - val_model
_acc: 0.8484
Epoch 25/60
3983/3983 [==============================] - 287s 72ms/step - loss: 0.0013 - model_acc: 0.9825 - val_loss: 0.0120 - val_model
_acc: 0.8487
Epoch 26/60
3983/3983 [==============================] - 281s 71ms/step - loss: 0.0012 - model_acc: 0.9834 - val_loss: 0.0120 - val_model
_acc: 0.8501
Epoch 27/60
3983/3983 [==============================] - 287s 72ms/step - loss: 0.0012 - model_acc: 0.9843 - val_loss: 0.0119 - val_model
_acc: 0.8473
Epoch 28/60
3983/3983 [==============================] - 284s 71ms/step - loss: 0.0011 - model_acc: 0.9852 - val_loss: 0.0118 - val_model
_acc: 0.8549
Epoch 29/60
3983/3983 [==============================] - 280s 70ms/step - loss: 0.0010 - model_acc: 0.9862 - val_loss: 0.0120 - val_model
_acc: 0.8481
Epoch 30/60
3983/3983 [==============================] - 281s 71ms/step - loss: 9.9793e-04 - model_acc: 0.9865 - val_loss: 0.0117 - val_m
odel_acc: 0.8527
Epoch 31/60
3983/3983 [==============================] - 278s 70ms/step - loss: 9.4331e-04 - model_acc: 0.9875 - val_loss: 0.0118 - val_m
odel_acc: 0.8526
Epoch 32/60
3983/3983 [==============================] - 284s 71ms/step - loss: 8.9778e-04 - model_acc: 0.9879 - val_loss: 0.0119 - val_m
```

```
odel_acc: 0.8526
Epoch 33/60
3983/3983 [==============================] - 280s 70ms/step - loss: 8.4628e-04 - model_acc: 0.9890 - val_loss: 0.0118 - val_m
odel_acc: 0.8521
Epoch 34/60
3983/3983 [==============================] - 284s 71ms/step - loss: 8.1440e-04 - model_acc: 0.9885 - val_loss: 0.0119 - val_m
odel_acc: 0.8481
Epoch 35/60
3983/3983 [==============================] - 289s 72ms/step - loss: 7.7626e-04 - model_acc: 0.9894 - val_loss: 0.0117 - val_m
odel_acc: 0.8555
Epoch 36/60
3983/3983 [==============================] - 278s 70ms/step - loss: 7.2963e-04 - model_acc: 0.9902 - val_loss: 0.0117 - val_m
odel_acc: 0.8490
Epoch 37/60
3983/3983 [==============================] - 284s 71ms/step - loss: 6.9979e-04 - model_acc: 0.9910 - val_loss: 0.0119 - val_m
odel_acc: 0.8532
Epoch 38/60
3983/3983 [==============================] - 281s 70ms/step - loss: 6.6430e-04 - model_acc: 0.9913 - val_loss: 0.0118 - val_m
odel_acc: 0.8504
Epoch 39/60
3983/3983 [==============================] - 285s 72ms/step - loss: 6.3398e-04 - model_acc: 0.9915 - val_loss: 0.0118 - val_m
odel_acc: 0.8552
Epoch 40/60
3983/3983 [==============================] - 284s 71ms/step - loss: 6.0754e-04 - model_acc: 0.9916 - val_loss: 0.0118 - val_m
odel_acc: 0.8504
Epoch 41/60
3983/3983 [==============================] - 274s 69ms/step - loss: 5.7828e-04 - model_acc: 0.9923 - val_loss: 0.0118 - val_m
odel_acc: 0.8521
Epoch 42/60
3983/3983 [==============================] - 280s 70ms/step - loss: 5.6146e-04 - model_acc: 0.9925 - val_loss: 0.0117 - val_m
odel_acc: 0.8501
Epoch 43/60
3983/3983 [==============================] - 277s 70ms/step - loss: 5.3389e-04 - model_acc: 0.9929 - val_loss: 0.0118 - val_m
odel_acc: 0.8566
Epoch 44/60
3983/3983 [==============================] - 277s 69ms/step - loss: 5.1579e-04 - model_acc: 0.9934 - val_loss: 0.0118 - val_m
odel_acc: 0.8535
Epoch 45/60
3983/3983 [==============================] - 279s 70ms/step - loss: 4.9054e-04 - model_acc: 0.9938 - val_loss: 0.0119 - val_m
odel_acc: 0.8470
Epoch 46/60
3983/3983 [==============================] - 275s 69ms/step - loss: 4.7542e-04 - model_acc: 0.9940 - val_loss: 0.0117 - val_m
odel_acc: 0.8501
Epoch 47/60
```

```
3983/3983 [==============================] - 280s 70ms/step - loss: 4.5706e-04 - model_acc: 0.9938 - val_loss: 0.0120 - val_m
odel_acc: 0.8521
Epoch 48/60
3983/3983 [==============================] - 277s 70ms/step - loss: 4.3540e-04 - model_acc: 0.9942 - val_loss: 0.0120 - val_m
odel_acc: 0.8493
Epoch 49/60
3983/3983 [==============================] - 277s 69ms/step - loss: 4.2509e-04 - model_acc: 0.9947 - val_loss: 0.0117 - val_m
odel_acc: 0.8504
Epoch 50/60
3983/3983 [==============================] - 279s 70ms/step - loss: 4.0464e-04 - model_acc: 0.9947 - val_loss: 0.0117 - val_m
odel_acc: 0.8518
Epoch 51/60
3983/3983 [==============================] - 275s 69ms/step - loss: 3.9917e-04 - model_acc: 0.9947 - val_loss: 0.0116 - val_m
odel_acc: 0.8521
Epoch 52/60
3983/3983 [==============================] - 280s 70ms/step - loss: 3.8722e-04 - model_acc: 0.9952 - val_loss: 0.0116 - val_m
odel_acc: 0.8566
Epoch 53/60
3983/3983 [==============================] - 277s 70ms/step - loss: 3.7402e-04 - model_acc: 0.9953 - val_loss: 0.0117 - val_m
odel_acc: 0.8501
Epoch 54/60
3983/3983 [==============================] - 277s 69ms/step - loss: 3.6542e-04 - model_acc: 0.9956 - val_loss: 0.0117 - val_m
odel_acc: 0.8487
Epoch 55/60
3983/3983 [==============================] - 279s 70ms/step - loss: 3.5140e-04 - model_acc: 0.9953 - val_loss: 0.0117 - val_m
odel_acc: 0.8512
Epoch 56/60
3983/3983 [==============================] - 274s 69ms/step - loss: 3.4467e-04 - model_acc: 0.9957 - val_loss: 0.0117 - val_m
odel_acc: 0.8538
Epoch 57/60
3983/3983 [==============================] - 279s 70ms/step - loss: 3.2958e-04 - model_acc: 0.9957 - val_loss: 0.0118 - val_m
odel_acc: 0.8490
Epoch 58/60
3983/3983 [==============================] - 277s 69ms/step - loss: 3.2438e-04 - model_acc: 0.9959 - val_loss: 0.0117 - val_m
odel_acc: 0.8473
Epoch 59/60
3983/3983 [==============================] - 276s 69ms/step - loss: 3.1895e-04 - model_acc: 0.9961 - val_loss: 0.0119 - val_m
odel_acc: 0.8481
Epoch 60/60
3983/3983 [==============================] - 278s 70ms/step - loss: 3.0951e-04 - model_acc: 0.9963 - val_loss: 0.0121 - val_m
odel_acc: 0.8453
```

```
Out[6]: <tensorflow.python.keras.callbacks.History at 0x27a00a4ca30>
```

```
In [ ]:
```