

```
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):
```

996/996 [=====] - 65s 65ms/step - loss: 5.1619e-04 - model_acc: 0.9929 - val_loss: 0.0138 - val_model_acc: 0.8264

Out[8]: <tensorflow.python.keras.callbacks.History at 0x2219e7690a0>

In [9]:

```
# VGG16 .2drop combined features
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("feature2", 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(0.2)(x)
x = Dense(units=4096, activation='relu')(x)
x = Dropout(0.2)(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, 2, 2, 512)
 (None, 512)
 feature6 (None, 512)
 combined feature (None, 1475)
 Model: "model_1"

Layer (type)	Output Shape	Param #	Connected to
=====			
input_2 (InputLayer)	[(None, 48, 48, 3)]	0	
conv2d_10 (Conv2D)	(None, 48, 48, 64)	1792	input_2[0][0]
conv2d_11 (Conv2D)	(None, 48, 48, 64)	36928	conv2d_10[0][0]
max_pooling2d_4 (MaxPooling2D)	(None, 24, 24, 64)	0	conv2d_11[0][0]
conv2d_12 (Conv2D)	(None, 24, 24, 128)	73856	max_pooling2d_4[0][0]
conv2d_13 (Conv2D)	(None, 24, 24, 128)	147584	conv2d_12[0][0]
max_pooling2d_5 (MaxPooling2D)	(None, 12, 12, 128)	0	conv2d_13[0][0]
conv2d_14 (Conv2D)	(None, 12, 12, 256)	295168	max_pooling2d_5[0][0]
conv2d_15 (Conv2D)	(None, 12, 12, 256)	590080	conv2d_14[0][0]
conv2d_16 (Conv2D)	(None, 12, 12, 256)	590080	conv2d_15[0][0]
max_pooling2d_6 (MaxPooling2D)	(None, 6, 6, 256)	0	conv2d_16[0][0]
conv2d_17 (Conv2D)	(None, 6, 6, 512)	1180160	max_pooling2d_6[0][0]
conv2d_18 (Conv2D)	(None, 6, 6, 512)	2359808	conv2d_17[0][0]
conv2d_19 (Conv2D)	(None, 6, 6, 512)	2359808	conv2d_18[0][0]
max_pooling2d_7 (MaxPooling2D)	(None, 3, 3, 512)	0	conv2d_19[0][0]
conv2d_20 (Conv2D)	(None, 3, 3, 512)	2359808	max_pooling2d_7[0][0]
conv2d_21 (Conv2D)	(None, 3, 3, 512)	2359808	conv2d_20[0][0]
conv2d_22 (Conv2D)	(None, 3, 3, 512)	2359808	conv2d_21[0][0]

max_pooling2d_8 (MaxPooling2D)	(None, 2, 2, 512)	0	conv2d_22[0][0]
global_average_pooling2d_6 (GlobalAveragePooling2D)	(None, 3)	0	input_2[0][0]
global_average_pooling2d_7 (GlobalAveragePooling2D)	(None, 64)	0	max_pooling2d_4[0][0]
global_average_pooling2d_8 (GlobalAveragePooling2D)	(None, 128)	0	max_pooling2d_5[0][0]
global_average_pooling2d_9 (GlobalAveragePooling2D)	(None, 256)	0	max_pooling2d_6[0][0]
global_average_pooling2d_10 (GlobalAveragePooling2D)	(None, 512)	0	max_pooling2d_7[0][0]
global_average_pooling2d_11 (GlobalAveragePooling2D)	(None, 512)	0	max_pooling2d_8[0][0]
tf.concat_1 (TFOpLambda)	(None, 1475)	0	global_average_pooling2d_6[0][0] global_average_pooling2d_7[0][0] global_average_pooling2d_8[0][0] global_average_pooling2d_9[0][0] global_average_pooling2d_10[0][0] global_average_pooling2d_11[0][0]
dense_3 (Dense)	(None, 4096)	6045696	tf.concat_1[0][0]
dropout_2 (Dropout)	(None, 4096)	0	dense_3[0][0]
dense_4 (Dense)	(None, 4096)	16781312	dropout_2[0][0]
dropout_3 (Dropout)	(None, 4096)	0	dense_4[0][0]
dense_5 (Dense)	(None, 8)	32776	dropout_3[0][0]

=====
Total params: 37,574,472

Trainable params: 37,574,472

Non-trainable params: 0

Epoch 1/30

996/996 [=====] - 73s 73ms/step - loss: 0.0583 - model_acc: 0.3634 - val_loss: 0.0550 - val_model_acc: 0.4191

Epoch 2/30

996/996 [=====] - 72s 73ms/step - loss: 0.0368 - model_acc: 0.6155 - val_loss: 0.0304 - val_model_acc: 0.6715

Epoch 3/30

996/996 [=====] - 74s 74ms/step - loss: 0.0257 - model_acc: 0.7145 - val_loss: 0.0252 - val_model_acc: 0.7182

Epoch 4/30
996/996 [=====] - 74s 74ms/step - loss: 0.0208 - model_acc: 0.7603 - val_loss: 0.0208 - val_model_acc: 0.7579

Epoch 5/30
996/996 [=====] - 72s 72ms/step - loss: 0.0174 - model_acc: 0.7958 - val_loss: 0.0198 - val_model_acc: 0.7666

Epoch 6/30
996/996 [=====] - 72s 72ms/step - loss: 0.0146 - model_acc: 0.8233 - val_loss: 0.0178 - val_model_acc: 0.7866

Epoch 7/30
996/996 [=====] - 72s 72ms/step - loss: 0.0124 - model_acc: 0.8478 - val_loss: 0.0181 - val_model_acc: 0.7906

Epoch 8/30
996/996 [=====] - 72s 72ms/step - loss: 0.0106 - model_acc: 0.8675 - val_loss: 0.0171 - val_model_acc: 0.7991

Epoch 9/30
996/996 [=====] - 72s 72ms/step - loss: 0.0092 - model_acc: 0.8830 - val_loss: 0.0161 - val_model_acc: 0.8047

Epoch 10/30
996/996 [=====] - 72s 72ms/step - loss: 0.0079 - model_acc: 0.8980 - val_loss: 0.0163 - val_model_acc: 0.8052

Epoch 11/30
996/996 [=====] - 72s 72ms/step - loss: 0.0070 - model_acc: 0.9124 - val_loss: 0.0167 - val_model_acc: 0.8002

Epoch 12/30
996/996 [=====] - 72s 72ms/step - loss: 0.0062 - model_acc: 0.9211 - val_loss: 0.0153 - val_model_acc: 0.8165

Epoch 13/30
996/996 [=====] - 76s 76ms/step - loss: 0.0055 - model_acc: 0.9299 - val_loss: 0.0152 - val_model_acc: 0.8187

Epoch 14/30
996/996 [=====] - 72s 72ms/step - loss: 0.0051 - model_acc: 0.9360 - val_loss: 0.0157 - val_model_acc: 0.8120

Epoch 15/30
996/996 [=====] - 72s 72ms/step - loss: 0.0046 - model_acc: 0.9405 - val_loss: 0.0148 - val_model_acc: 0.8228

Epoch 16/30
996/996 [=====] - 72s 72ms/step - loss: 0.0043 - model_acc: 0.9420 - val_loss: 0.0147 - val_model_acc: 0.8219

Epoch 17/30
996/996 [=====] - 72s 72ms/step - loss: 0.0039 - model_acc: 0.9483 - val_loss: 0.0145 - val_model_acc: 0.8196

Epoch 18/30
996/996 [=====] - 72s 72ms/step - loss: 0.0037 - model_acc: 0.9525 - val_loss: 0.0152 - val_model_acc: 0.

```
8171
Epoch 19/30
996/996 [=====] - 72s 72ms/step - loss: 0.0034 - model_acc: 0.9552 - val_loss: 0.0146 - val_model_acc: 0.
8207
Epoch 20/30
996/996 [=====] - 72s 72ms/step - loss: 0.0032 - model_acc: 0.9591 - val_loss: 0.0147 - val_model_acc: 0.
8111
Epoch 21/30
996/996 [=====] - 72s 72ms/step - loss: 0.0032 - model_acc: 0.9572 - val_loss: 0.0149 - val_model_acc: 0.
8208
Epoch 22/30
996/996 [=====] - 72s 72ms/step - loss: 0.0030 - model_acc: 0.9598 - val_loss: 0.0143 - val_model_acc: 0.
8183
Epoch 23/30
996/996 [=====] - 72s 72ms/step - loss: 0.0028 - model_acc: 0.9633 - val_loss: 0.0143 - val_model_acc: 0.
8187
Epoch 24/30
996/996 [=====] - 72s 72ms/step - loss: 0.0026 - model_acc: 0.9646 - val_loss: 0.0146 - val_model_acc: 0.
8199
Epoch 25/30
996/996 [=====] - 72s 72ms/step - loss: 0.0025 - model_acc: 0.9655 - val_loss: 0.0142 - val_model_acc: 0.
8224
Epoch 26/30
996/996 [=====] - 72s 72ms/step - loss: 0.0023 - model_acc: 0.9675 - val_loss: 0.0145 - val_model_acc: 0.
8185
Epoch 27/30
996/996 [=====] - 72s 72ms/step - loss: 0.0023 - model_acc: 0.9688 - val_loss: 0.0141 - val_model_acc: 0.
8250
Epoch 28/30
996/996 [=====] - 72s 72ms/step - loss: 0.0022 - model_acc: 0.9690 - val_loss: 0.0143 - val_model_acc: 0.
8202
Epoch 29/30
996/996 [=====] - 72s 72ms/step - loss: 0.0021 - model_acc: 0.9730 - val_loss: 0.0142 - val_model_acc: 0.
8212
Epoch 30/30
996/996 [=====] - 72s 72ms/step - loss: 0.0020 - model_acc: 0.9723 - val_loss: 0.0141 - val_model_acc: 0.
8202
Epoch 1/30
996/996 [=====] - 72s 72ms/step - loss: 0.0016 - model_acc: 0.9805 - val_loss: 0.0134 - val_model_acc: 0.
8278
Epoch 2/30
996/996 [=====] - 72s 72ms/step - loss: 0.0012 - model_acc: 0.9896 - val_loss: 0.0134 - val_model_acc: 0.
8227
Epoch 3/30
```



```
996/996 [=====] - 72s 72ms/step - loss: 0.0011 - model_acc: 0.9916 - val_loss: 0.0134 - val_model_acc: 0.8275
Epoch 4/30
996/996 [=====] - 72s 72ms/step - loss: 0.0010 - model_acc: 0.9909 - val_loss: 0.0135 - val_model_acc: 0.8264
Epoch 5/30
996/996 [=====] - 72s 72ms/step - loss: 9.9372e-04 - model_acc: 0.9908 - val_loss: 0.0135 - val_model_acc: 0.8233
Epoch 6/30
996/996 [=====] - 72s 72ms/step - loss: 9.7162e-04 - model_acc: 0.9907 - val_loss: 0.0135 - val_model_acc: 0.8235
Epoch 7/30
996/996 [=====] - 72s 72ms/step - loss: 9.0662e-04 - model_acc: 0.9921 - val_loss: 0.0135 - val_model_acc: 0.8269
Epoch 8/30
996/996 [=====] - 72s 72ms/step - loss: 8.5687e-04 - model_acc: 0.9917 - val_loss: 0.0136 - val_model_acc: 0.8247
Epoch 9/30
996/996 [=====] - 72s 72ms/step - loss: 8.3074e-04 - model_acc: 0.9919 - val_loss: 0.0135 - val_model_acc: 0.8247
Epoch 10/30
996/996 [=====] - 72s 72ms/step - loss: 7.9006e-04 - model_acc: 0.9925 - val_loss: 0.0137 - val_model_acc: 0.8272
Epoch 11/30
996/996 [=====] - 72s 72ms/step - loss: 7.8091e-04 - model_acc: 0.9917 - val_loss: 0.0135 - val_model_acc: 0.8261
Epoch 12/30
996/996 [=====] - 72s 72ms/step - loss: 7.4782e-04 - model_acc: 0.9931 - val_loss: 0.0136 - val_model_acc: 0.8278
Epoch 13/30
996/996 [=====] - 72s 72ms/step - loss: 7.0897e-04 - model_acc: 0.9932 - val_loss: 0.0134 - val_model_acc: 0.8294
Epoch 14/30
996/996 [=====] - 72s 72ms/step - loss: 6.8994e-04 - model_acc: 0.9926 - val_loss: 0.0136 - val_model_acc: 0.8264
Epoch 15/30
996/996 [=====] - 72s 72ms/step - loss: 6.6829e-04 - model_acc: 0.9936 - val_loss: 0.0135 - val_model_acc: 0.8289
Epoch 16/30
996/996 [=====] - 72s 72ms/step - loss: 6.3858e-04 - model_acc: 0.9940 - val_loss: 0.0136 - val_model_acc: 0.8289
Epoch 17/30
996/996 [=====] - 72s 72ms/step - loss: 6.2588e-04 - model_acc: 0.9930 - val_loss: 0.0135 - val_model_acc: 0.8230
```

```

Epoch 18/30
996/996 [=====] - 72s 72ms/step - loss: 6.0470e-04 - model_acc: 0.9941 - val_loss: 0.0135 - val_model_acc: 0.8317
Epoch 19/30
996/996 [=====] - 72s 72ms/step - loss: 5.8416e-04 - model_acc: 0.9944 - val_loss: 0.0133 - val_model_acc: 0.8306
Epoch 20/30
996/996 [=====] - 72s 72ms/step - loss: 5.6881e-04 - model_acc: 0.9935 - val_loss: 0.0134 - val_model_acc: 0.8320
Epoch 21/30
996/996 [=====] - 72s 72ms/step - loss: 5.5340e-04 - model_acc: 0.9937 - val_loss: 0.0136 - val_model_acc: 0.8292
Epoch 22/30
996/996 [=====] - 72s 72ms/step - loss: 5.4054e-04 - model_acc: 0.9942 - val_loss: 0.0135 - val_model_acc: 0.8264
Epoch 23/30
996/996 [=====] - 72s 72ms/step - loss: 5.3157e-04 - model_acc: 0.9939 - val_loss: 0.0136 - val_model_acc: 0.8306
Epoch 24/30
996/996 [=====] - 72s 72ms/step - loss: 5.0337e-04 - model_acc: 0.9951 - val_loss: 0.0135 - val_model_acc: 0.8264
Epoch 25/30
996/996 [=====] - 72s 72ms/step - loss: 4.9640e-04 - model_acc: 0.9944 - val_loss: 0.0133 - val_model_acc: 0.8311
Epoch 26/30
996/996 [=====] - 72s 72ms/step - loss: 4.9555e-04 - model_acc: 0.9945 - val_loss: 0.0135 - val_model_acc: 0.8314
Epoch 27/30
996/996 [=====] - 72s 72ms/step - loss: 4.8011e-04 - model_acc: 0.9946 - val_loss: 0.0135 - val_model_acc: 0.8314
Epoch 28/30
996/996 [=====] - 72s 72ms/step - loss: 4.6740e-04 - model_acc: 0.9951 - val_loss: 0.0135 - val_model_acc: 0.8306
Epoch 29/30
996/996 [=====] - 72s 72ms/step - loss: 4.5285e-04 - model_acc: 0.9948 - val_loss: 0.0135 - val_model_acc: 0.8292
Epoch 30/30
996/996 [=====] - 72s 72ms/step - loss: 4.4600e-04 - model_acc: 0.9949 - val_loss: 0.0134 - val_model_acc: 0.8280

```

Out[9]: <tensorflow.python.keras.callbacks.History at 0x224c395b2b0>

In [10]: *# VGG16 no drop/combine*