

```
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 [12]: 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 [13]: 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=False
model = Sequential([
    base_model,
    layers.GlobalAveragePooling2D(),
    layers.Dense(2048, activation='relu'),
    layers.Dense(2048, activation='relu'),
    layers.Dense(2048, activation='relu'),
    layers.Dense(emotions_count, activation='softmax'),
])

model.compile(optimizer=adam.Adam(learning_rate=5e-4),
              loss=losses.CategoricalCrossentropy(),
              metrics = [model_acc])

model.fit(x=training_images,
          y=training_emotions,
          batch_size=8,
          epochs=30,
          validation_data=(test_images, test_emotions))
```

C:\Users\Dark1\anaconda3\lib\site-packages\tensorflow\python\data\ops\dataset_ops.py:3703: UserWarning: Even though the `tf.config.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

3983/3983 [=====] - 146s 36ms/step - loss: 1.3778 - model_acc: 0.4993 - val_loss: 1.3177 - val_model_acc: 0.5383

Epoch 2/30

3983/3983 [=====] - 138s 35ms/step - loss: 1.2452 - model_acc: 0.5580 - val_loss: 1.2617 - val_model_acc: 0.5538

Epoch 3/30

3983/3983 [=====] - 138s 35ms/step - loss: 1.1642 - model_acc: 0.5887 - val_loss: 1.2767 - val_model_acc: 0.5476

Epoch 4/30

3983/3983 [=====] - 142s 36ms/step - loss: 1.0962 - model_acc: 0.6171 - val_loss: 1.1902 - val_model_acc: 0.5863

Epoch 5/30

```
3983/3983 [=====] - 144s 36ms/step - loss: 1.0225 - model_acc: 0.6389 - val_loss: 1.2584 - val_model_acc: 0.5818
Epoch 6/30
3983/3983 [=====] - 141s 35ms/step - loss: 0.9502 - model_acc: 0.6690 - val_loss: 1.3156 - val_model_acc: 0.5784
Epoch 7/30
3983/3983 [=====] - 144s 36ms/step - loss: 0.8757 - model_acc: 0.6953 - val_loss: 1.2834 - val_model_acc: 0.5934
Epoch 8/30
3983/3983 [=====] - 142s 36ms/step - loss: 0.8059 - model_acc: 0.7214 - val_loss: 1.3253 - val_model_acc: 0.5809
Epoch 9/30
3983/3983 [=====] - 142s 36ms/step - loss: 0.7317 - model_acc: 0.7481 - val_loss: 1.4267 - val_model_acc: 0.5722
Epoch 10/30
3983/3983 [=====] - 144s 36ms/step - loss: 0.6618 - model_acc: 0.7755 - val_loss: 1.5284 - val_model_acc: 0.5985
Epoch 11/30
3983/3983 [=====] - 142s 36ms/step - loss: 0.5971 - model_acc: 0.7980 - val_loss: 1.5658 - val_model_acc: 0.5852
Epoch 12/30
3983/3983 [=====] - 143s 36ms/step - loss: 0.5394 - model_acc: 0.8221 - val_loss: 1.7137 - val_model_acc: 0.5999
Epoch 13/30
3983/3983 [=====] - 142s 36ms/step - loss: 0.4894 - model_acc: 0.8413 - val_loss: 1.7915 - val_model_acc: 0.5860
Epoch 14/30
3983/3983 [=====] - 143s 36ms/step - loss: 0.4435 - model_acc: 0.8589 - val_loss: 1.8987 - val_model_acc: 0.6033
Epoch 15/30
3983/3983 [=====] - 142s 36ms/step - loss: 0.4004 - model_acc: 0.8766 - val_loss: 2.1475 - val_model_acc: 0.5697
Epoch 16/30
3983/3983 [=====] - 143s 36ms/step - loss: 0.3719 - model_acc: 0.8849 - val_loss: 2.3597 - val_model_acc: 0.5875
Epoch 17/30
3983/3983 [=====] - 145s 37ms/step - loss: 0.3369 - model_acc: 0.9023 - val_loss: 2.3149 - val_model_acc: 0.6042
Epoch 18/30
3983/3983 [=====] - 144s 36ms/step - loss: 0.3180 - model_acc: 0.9094 - val_loss: 2.1468 - val_model_acc: 0.5928
Epoch 19/30
3983/3983 [=====] - 146s 37ms/step - loss: 0.2960 - model_acc: 0.9169 - val_loss: 2.2266 - val_model_acc: 0.5895
Epoch 20/30
3983/3983 [=====] - 145s 36ms/step - loss: 0.2709 - model_acc: 0.9254 - val_loss: 2.4519 - val_model_acc: 0.6042
Epoch 21/30
3983/3983 [=====] - 144s 36ms/step - loss: 0.2607 - model_acc: 0.9288 - val_loss: 2.6910 - val_model_acc: 0.6068
Epoch 22/30
3983/3983 [=====] - 143s 36ms/step - loss: 0.2458 - model_acc: 0.9363 - val_loss: 2.7299 - val_model_acc: 0.5968
Epoch 23/30
3983/3983 [=====] - 143s 36ms/step - loss: 0.2407 - model_acc: 0.9387 - val_loss: 2.9007 - val_model_acc: 0.5960
Epoch 24/30
3983/3983 [=====] - 143s 36ms/step - loss: 0.2306 - model_acc: 0.9428 - val_loss: 3.0258 - val_model_acc: 0.6010
Epoch 25/30
3983/3983 [=====] - 144s 36ms/step - loss: 0.2195 - model_acc: 0.9455 - val_loss: 2.9103 - val_model_acc: 0.5816
Epoch 26/30
3983/3983 [=====] - 142s 36ms/step - loss: 0.2145 - model_acc: 0.9495 - val_loss: 3.1620 - val_model_acc: 0.6107
```

Epoch 27/30

3983/3983 [=====] - 146s 37ms/step - loss: 0.1983 - model_acc: 0.9544 - val_loss: 2.9478 - val_model_acc: 0.5983

Epoch 28/30

3983/3983 [=====] - 144s 36ms/step - loss: 0.1930 - model_acc: 0.9558 - val_loss: 3.1247 - val_model_acc: 0.5966

Epoch 29/30

3983/3983 [=====] - 143s 36ms/step - loss: 0.1907 - model_acc: 0.9564 - val_loss: 3.2111 - val_model_acc: 0.5957

Epoch 30/30

3983/3983 [=====] - 143s 36ms/step - loss: 0.1964 - model_acc: 0.9550 - val_loss: 3.1740 - val_model_acc: 0.6036

Out[5]: <tensorflow.python.keras.callbacks.History at 0x1d9f4846040>

```
In [6]: base_model = vgg16.VGG16(include_top=False,
                                weights="imagenet",
                                input_shape=(48,48,3))

base_model.trainable=False
model = Sequential([
    base_model,
    layers.GlobalAveragePooling2D(),
    layers.Dense(4096, activation='relu'),
    layers.Dense(emotions_count, activation='softmax'),
])

model.compile(optimizer=adam.Adam(learning_rate=1e-4),
              loss='mse',
              metrics = [model_acc])

model.fit(x=training_images,
          y=training_emotions,
          batch_size=32,
          epochs=20,
          validation_data=(test_images, test_emotions))
```

Epoch 1/20

996/996 [=====] - 67s 66ms/step - loss: 0.0790 - model_acc: 0.5076 - val_loss: 0.0768 - val_model_acc: 0.5330

Epoch 2/20

996/996 [=====] - 65s 65ms/step - loss: 0.0735 - model_acc: 0.5544 - val_loss: 0.0743 - val_model_acc: 0.5473

Epoch 3/20

996/996 [=====] - 66s 66ms/step - loss: 0.0705 - model_acc: 0.5767 - val_loss: 0.0731 - val_model_acc: 0.5589

Epoch 4/20

996/996 [=====] - 67s 68ms/step - loss: 0.0682 - model_acc: 0.5926 - val_loss: 0.0728 - val_model_acc: 0.5605

Epoch 5/20

996/996 [=====] - 68s 69ms/step - loss: 0.0658 - model_acc: 0.6151 - val_loss: 0.0717 - val_model_acc: 0.5639

Epoch 6/20

996/996 [=====] - 66s 67ms/step - loss: 0.0637 - model_acc: 0.6294 - val_loss: 0.0701 - val_model_acc: 0.5754

Epoch 7/20

996/996 [=====] - 65s 65ms/step - loss: 0.0615 - model_acc: 0.6460 - val_loss: 0.0699 - val_model_acc: 0.5794

Epoch 8/20

996/996 [=====] - 66s 66ms/step - loss: 0.0594 - model_acc: 0.6604 - val_loss: 0.0700 - val_model_acc: 0.5732

Epoch 9/20

996/996 [=====] - 65s 65ms/step - loss: 0.0573 - model_acc: 0.6777 - val_loss: 0.0689 - val_model_acc: 0.5886

Epoch 10/20

996/996 [=====] - 65s 66ms/step - loss: 0.0554 - model_acc: 0.6927 - val_loss: 0.0700 - val_model_acc: 0.5788

Epoch 11/20

996/996 [=====] - 65s 65ms/step - loss: 0.0534 - model_acc: 0.7058 - val_loss: 0.0695 - val_model_acc: 0.5862

```
Epoch 12/20
996/996 [=====] - 65s 66ms/step - loss: 0.0514 - model_acc: 0.7208 - val_loss: 0.0676 - val_model_acc: 0.5960
Epoch 13/20
996/996 [=====] - 65s 65ms/step - loss: 0.0496 - model_acc: 0.7335 - val_loss: 0.0678 - val_model_acc: 0.5951
Epoch 14/20
996/996 [=====] - 65s 65ms/step - loss: 0.0478 - model_acc: 0.7461 - val_loss: 0.0675 - val_model_acc: 0.6027
Epoch 15/20
996/996 [=====] - 65s 65ms/step - loss: 0.0459 - model_acc: 0.7600 - val_loss: 0.0677 - val_model_acc: 0.6027
Epoch 16/20
996/996 [=====] - 66s 66ms/step - loss: 0.0442 - model_acc: 0.7723 - val_loss: 0.0678 - val_model_acc: 0.5994
Epoch 17/20
996/996 [=====] - 66s 66ms/step - loss: 0.0426 - model_acc: 0.7832 - val_loss: 0.0681 - val_model_acc: 0.6016
Epoch 18/20
996/996 [=====] - 67s 67ms/step - loss: 0.0412 - model_acc: 0.7911 - val_loss: 0.0676 - val_model_acc: 0.6128
Epoch 19/20
996/996 [=====] - 65s 65ms/step - loss: 0.0395 - model_acc: 0.8013 - val_loss: 0.0683 - val_model_acc: 0.5900
Epoch 20/20
996/996 [=====] - 65s 65ms/step - loss: 0.0379 - model_acc: 0.8125 - val_loss: 0.0674 - val_model_acc: 0.6084
```

Out[6]: <tensorflow.python.keras.callbacks.History at 0x1dc4e171d30>

```
In [7]: base_model = vgg16.VGG16(include_top=False,
                                weights="imagenet",
                                input_shape=(48,48,3))

base_model.trainable=False
model = Sequential([
    base_model,
    layers.GlobalAveragePooling2D(),
    layers.Dense(4096, activation='relu'),
    layers.Dense(emotions_count, activation='softmax'),
])

model.compile(optimizer='sgd',
              loss='mse',
              metrics = [model_acc])

model.fit(x=training_images,
          y=training_emotions,
          batch_size=32,
          epochs=20,
          validation_data=(test_images, test_emotions))
```

Epoch 1/20

996/996 [=====] - 65s 65ms/step - loss: 0.0916 - model_acc: 0.3873 - val_loss: 0.0894 - val_model_acc: 0.4251

Epoch 2/20

996/996 [=====] - 66s 66ms/step - loss: 0.0870 - model_acc: 0.4497 - val_loss: 0.0869 - val_model_acc: 0.4330

Epoch 3/20

996/996 [=====] - 65s 65ms/step - loss: 0.0847 - model_acc: 0.4695 - val_loss: 0.0851 - val_model_acc: 0.4674

Epoch 4/20

996/996 [=====] - 65s 65ms/step - loss: 0.0834 - model_acc: 0.4811 - val_loss: 0.0840 - val_model_acc: 0.4741

Epoch 5/20

996/996 [=====] - 65s 65ms/step - loss: 0.0824 - model_acc: 0.4882 - val_loss: 0.0832 - val_model_acc: 0.4839

Epoch 6/20

996/996 [=====] - 64s 65ms/step - loss: 0.0816 - model_acc: 0.4935 - val_loss: 0.0825 - val_model_acc: 0.4870

Epoch 7/20

996/996 [=====] - 66s 66ms/step - loss: 0.0810 - model_acc: 0.4981 - val_loss: 0.0821 - val_model_acc: 0.4944

Epoch 8/20

996/996 [=====] - 65s 66ms/step - loss: 0.0805 - model_acc: 0.5008 - val_loss: 0.0816 - val_model_acc: 0.5011

Epoch 9/20

996/996 [=====] - 66s 67ms/step - loss: 0.0801 - model_acc: 0.5060 - val_loss: 0.0813 - val_model_acc: 0.4997

Epoch 10/20

996/996 [=====] - 65s 65ms/step - loss: 0.0797 - model_acc: 0.5099 - val_loss: 0.0809 - val_model_acc: 0.5032

Epoch 11/20

996/996 [=====] - 66s 67ms/step - loss: 0.0793 - model_acc: 0.5110 - val_loss: 0.0806 - val_model_acc: 0.5068


```
Epoch 12/20
996/996 [=====] - 65s 65ms/step - loss: 0.0790 - model_acc: 0.5147 - val_loss: 0.0804 - val_model_acc: 0.5065
Epoch 13/20
996/996 [=====] - 62s 63ms/step - loss: 0.0787 - model_acc: 0.5168 - val_loss: 0.0801 - val_model_acc: 0.5096
Epoch 14/20
996/996 [=====] - 62s 63ms/step - loss: 0.0784 - model_acc: 0.5173 - val_loss: 0.0799 - val_model_acc: 0.5085
Epoch 15/20
996/996 [=====] - 62s 63ms/step - loss: 0.0782 - model_acc: 0.5202 - val_loss: 0.0797 - val_model_acc: 0.5085
Epoch 16/20
996/996 [=====] - 63s 63ms/step - loss: 0.0780 - model_acc: 0.5223 - val_loss: 0.0795 - val_model_acc: 0.5136
Epoch 17/20
996/996 [=====] - 63s 63ms/step - loss: 0.0778 - model_acc: 0.5218 - val_loss: 0.0792 - val_model_acc: 0.5161
Epoch 18/20
996/996 [=====] - 65s 66ms/step - loss: 0.0775 - model_acc: 0.5250 - val_loss: 0.0792 - val_model_acc: 0.5178
Epoch 19/20
996/996 [=====] - 64s 65ms/step - loss: 0.0774 - model_acc: 0.5272 - val_loss: 0.0790 - val_model_acc: 0.5164
Epoch 20/20
996/996 [=====] - 64s 65ms/step - loss: 0.0772 - model_acc: 0.5282 - val_loss: 0.0789 - val_model_acc: 0.5178
```

```
Out[7]: <tensorflow.python.keras.callbacks.History at 0x1d9f5916bb0>
```

```
In [*]: base_model = resnet.ResNet50(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))
```

Epoch 1/25

996/996 [=====] - 164s 164ms/step - loss: 1.0580 - model_acc: 0.6451 - val_loss: 0.9705 - val_model_acc: 0.6833

Epoch 2/25

996/996 [=====] - 163s 164ms/step - loss: 0.6839 - model_acc: 0.7829 - val_loss: 0.7552 - val_model_acc: 0.7587

Epoch 3/25

996/996 [=====] - 164s 164ms/step - loss: 0.5033 - model_acc: 0.8446 - val_loss: 0.7490 - val_model_acc: 0.7662

Epoch 4/25

996/996 [=====] - 164s 165ms/step - loss: 0.3663 - model_acc: 0.8935 - val_loss: 0.7871 - val_model_acc: 0.7586

Epoch 5/25

996/996 [=====] - 166s 167ms/step - loss: 0.2751 - model_acc: 0.9284 - val_loss: 0.9031 - val_model_acc: 0.7599

Epoch 6/25

996/996 [=====] - 164s 165ms/step - loss: 0.2257 - model_acc: 0.9456 - val_loss: 0.9020 - val_model_acc: 0.7661

Epoch 7/25

996/996 [=====] - 165s 165ms/step - loss: 0.1872 - model_acc: 0.9574 - val_loss: 0.9195 - val_model_acc: 0.7735

Epoch 8/25

996/996 [=====] - 164s 165ms/step - loss: 0.1740 - model_acc: 0.9630 - val_loss: 0.9074 - val_model_acc: 0.7813

Epoch 9/25

996/996 [=====] - 160s 161ms/step - loss: 0.1548 - model_acc: 0.9702 - val_loss: 0.9542 - val_model_acc: 0.7642

Epoch 10/25

996/996 [=====] - 154s 155ms/step - loss: 0.1470 - model_acc: 0.9731 - val_loss: 1.0538 - val_model_acc: 0.7724

Epoch 11/25

```
996/996 [=====] - 153s 154ms/step - loss: 0.1375 - model_acc: 0.9752 - val_loss: 1.0872 - val_model_acc: 0.7825
Epoch 12/25
996/996 [=====] - 161s 162ms/step - loss: 0.1315 - model_acc: 0.9764 - val_loss: 1.0946 - val_model_acc: 0.7692
Epoch 13/25
465/996 [=====>.....] - ETA: 1:23 - loss: 0.1169 - model_acc: 0.9835
```

```
In [*]: 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))
```

```
In [ ]:
```