```python
from __future__ import print_function
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Activation,
Flatten, BatchNormalization,Conv2D, MaxPooling2D
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint

num_classes = 7
img_rows,img_cols = 48,48
batch_size = 32

train_data_dir= r'C:\FER-2013\train'
validation_data_dir= r'C:\FER-2013\test'

train_datagen = ImageDataGenerator(
                            rescale=1./255,
                            rotation_range=30,
                            shear_range=0.3,
                            zoom_range=0.3,
                            width_shift_range=0.4,
                            height_shift_range=0.4,
                            horizontal_flip=True,
                            fill_mode='nearest')

validation_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
                            train_data_dir,
                            color_mode='grayscale',
                            target_size=(img_rows,img_cols),
                            batch_size=batch_size,
                            class_mode='categorical',
                            shuffle=True)

validation_generator = validation_datagen.flow_from_directory(
                                       validation_data_dir,
                                       color_mode='grayscale',
                                       target_size=(img_rows,img_cols),
                                       batch_size=batch_size,
                                       class_mode='categorical',
                                       shuffle=True)
```

```
Found 28709 images belonging to 7 classes.
Found 7178 images belonging to 7 classes.
```

```python
model = Sequential()

# Block-1

model.add(Conv2D(32,
(3,3),padding='same',kernel_initializer='he_normal',input_shape=(img_r
ows,img_cols,1)))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Conv2D(32,
(3,3),padding='same',kernel_initializer='he_normal',input_shape=(img_r
ows,img_cols,1)))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

# Block-2

model.add(Conv2D(64,
(3,3),padding='same',kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Conv2D(64,
(3,3),padding='same',kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

# Block-3

model.add(Conv2D(128,
(3,3),padding='same',kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Conv2D(128,
(3,3),padding='same',kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

# Block-4

model.add(Conv2D(256,
(3,3),padding='same',kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Conv2D(256,
(3,3),padding='same',kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))
```

```python
# Block-5

model.add(Flatten())
model.add(Dense(64,kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))

# Block-6

model.add(Dense(64,kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))

# Block-7

model.add(Dense(num_classes,kernel_initializer='he_normal'))
model.add(Activation('softmax'))

print(model.summary())
```

```
C:\Users\Asus\AppData\Roaming\Python\Python312\site-packages\keras\
src\layers\convolutional\base_conv.py:107: UserWarning: Do not pass an
`input_shape`/`input_dim` argument to a layer. When using Sequential
models, prefer using an `Input(shape)` object as the first layer in
the model instead.
  super().__init__(activity_regularizer=activity_regularizer,
**kwargs)

Model: "sequential"

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━┓
━━━━━━━━━━━━━━━━━┓
┃ Layer (type)                         ┃ Output Shape               ┃
Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━┩
━━━━━━━━━━━━━━━━━┩
│ conv2d (Conv2D)                      │ (None, 48, 48, 32)         │
320 │
├──────────────────────────────────────┼────────────────────────────┤
─────────────────┤
│ activation (Activation)              │ (None, 48, 48, 32)         │
0 │
├──────────────────────────────────────┼────────────────────────────┤
─────────────────┤
│ batch_normalization                  │ (None, 48, 48, 32)         │
128 │
│ (BatchNormalization)                 │                            │
│
```

```
├───────────────────────────────────────┼────────────────────────────────┤
─────────────────┤
│ conv2d_1 (Conv2D)                     │ (None, 48, 48, 32)             │
9,248 │
├───────────────────────────────────────┼────────────────────────────────┤
─────────────────┤
│ activation_1 (Activation)             │ (None, 48, 48, 32)             │
0 │
├───────────────────────────────────────┼────────────────────────────────┤
─────────────────┤
│ batch_normalization_1                 │ (None, 48, 48, 32)             │
128 │
│ (BatchNormalization)                  │                                │
│
├───────────────────────────────────────┼────────────────────────────────┤
─────────────────┤
│ max_pooling2d (MaxPooling2D)          │ (None, 24, 24, 32)             │
0 │
├───────────────────────────────────────┼────────────────────────────────┤
─────────────────┤
│ dropout (Dropout)                     │ (None, 24, 24, 32)             │
0 │
├───────────────────────────────────────┼────────────────────────────────┤
─────────────────┤
│ conv2d_2 (Conv2D)                     │ (None, 24, 24, 64)             │
18,496 │
├───────────────────────────────────────┼────────────────────────────────┤
─────────────────┤
│ activation_2 (Activation)             │ (None, 24, 24, 64)             │
0 │
├───────────────────────────────────────┼────────────────────────────────┤
─────────────────┤
│ batch_normalization_2                 │ (None, 24, 24, 64)             │
256 │
│ (BatchNormalization)                  │                                │
│
├───────────────────────────────────────┼────────────────────────────────┤
─────────────────┤
│ conv2d_3 (Conv2D)                     │ (None, 24, 24, 64)             │
36,928 │
├───────────────────────────────────────┼────────────────────────────────┤
─────────────────┤
│ activation_3 (Activation)             │ (None, 24, 24, 64)             │
0 │
├───────────────────────────────────────┼────────────────────────────────┤
─────────────────┤
│ batch_normalization_3                 │ (None, 24, 24, 64)             │
256 │
│ (BatchNormalization)                  │                                │
```

```
│
├─────────────────────────────────────┼─────────────────────────────┼
────────────────┤
│ max_pooling2d_1 (MaxPooling2D)      │ (None, 12, 12, 64)          │
0 │
├─────────────────────────────────────┼─────────────────────────────┼
────────────────┤
│ dropout_1 (Dropout)                 │ (None, 12, 12, 64)          │
0 │
├─────────────────────────────────────┼─────────────────────────────┼
────────────────┤
│ conv2d_4 (Conv2D)                   │ (None, 12, 12, 128)         │
73,856 │
├─────────────────────────────────────┼─────────────────────────────┼
────────────────┤
│ activation_4 (Activation)           │ (None, 12, 12, 128)         │
0 │
├─────────────────────────────────────┼─────────────────────────────┼
────────────────┤
│ batch_normalization_4               │ (None, 12, 12, 128)         │
512 │
│ (BatchNormalization)                │                             │
│
├─────────────────────────────────────┼─────────────────────────────┼
────────────────┤
│ conv2d_5 (Conv2D)                   │ (None, 12, 12, 128)         │
147,584 │
├─────────────────────────────────────┼─────────────────────────────┼
────────────────┤
│ activation_5 (Activation)           │ (None, 12, 12, 128)         │
0 │
├─────────────────────────────────────┼─────────────────────────────┼
────────────────┤
│ batch_normalization_5               │ (None, 12, 12, 128)         │
512 │
│ (BatchNormalization)                │                             │
│
├─────────────────────────────────────┼─────────────────────────────┼
────────────────┤
│ max_pooling2d_2 (MaxPooling2D)      │ (None, 6, 6, 128)           │
0 │
├─────────────────────────────────────┼─────────────────────────────┼
────────────────┤
│ dropout_2 (Dropout)                 │ (None, 6, 6, 128)           │
0 │
├─────────────────────────────────────┼─────────────────────────────┼
────────────────┤
│ conv2d_6 (Conv2D)                   │ (None, 6, 6, 256)           │
295,168 │
```

```
├─────────────────────────────────────────┼────────────────────────────────┼
──────────────────┤
│ activation_6 (Activation)               │ (None, 6, 6, 256)              │
0 │
├─────────────────────────────────────────┼────────────────────────────────┼
──────────────────┤
│ batch_normalization_6                   │ (None, 6, 6, 256)              │
1,024 │
│ (BatchNormalization)                    │                                │
│
├─────────────────────────────────────────┼────────────────────────────────┼
──────────────────┤
│ conv2d_7 (Conv2D)                       │ (None, 6, 6, 256)              │
590,080 │
├─────────────────────────────────────────┼────────────────────────────────┼
──────────────────┤
│ activation_7 (Activation)               │ (None, 6, 6, 256)              │
0 │
├─────────────────────────────────────────┼────────────────────────────────┼
──────────────────┤
│ batch_normalization_7                   │ (None, 6, 6, 256)              │
1,024 │
│ (BatchNormalization)                    │                                │
│
├─────────────────────────────────────────┼────────────────────────────────┼
──────────────────┤
│ max_pooling2d_3 (MaxPooling2D)          │ (None, 3, 3, 256)              │
0 │
├─────────────────────────────────────────┼────────────────────────────────┼
──────────────────┤
│ dropout_3 (Dropout)                     │ (None, 3, 3, 256)              │
0 │
├─────────────────────────────────────────┼────────────────────────────────┼
──────────────────┤
│ flatten (Flatten)                       │ (None, 2304)                   │
0 │
├─────────────────────────────────────────┼────────────────────────────────┼
──────────────────┤
│ dense (Dense)                           │ (None, 64)                     │
147,520 │
├─────────────────────────────────────────┼────────────────────────────────┼
──────────────────┤
│ activation_8 (Activation)               │ (None, 64)                     │
0 │
├─────────────────────────────────────────┼────────────────────────────────┼
──────────────────┤
│ batch_normalization_8                   │ (None, 64)                     │
256 │
│ (BatchNormalization)                    │                                │
```

```
│
├──────────────────────────────────────┼─────────────────────────────┼
─────────────────┤
│ dropout_4 (Dropout)                  │ (None, 64)                  │
0 │
├──────────────────────────────────────┼─────────────────────────────┼
─────────────────┤
│ dense_1 (Dense)                      │ (None, 64)                  │
4,160 │
├──────────────────────────────────────┼─────────────────────────────┼
─────────────────┤
│ activation_9 (Activation)            │ (None, 64)                  │
0 │
├──────────────────────────────────────┼─────────────────────────────┼
─────────────────┤
│ batch_normalization_9                │ (None, 64)                  │
256 │
│ (BatchNormalization)                 │                             │
│
├──────────────────────────────────────┼─────────────────────────────┼
─────────────────┤
│ dropout_5 (Dropout)                  │ (None, 64)                  │
0 │
├──────────────────────────────────────┼─────────────────────────────┼
─────────────────┤
│ dense_2 (Dense)                      │ (None, 7)                   │
455 │
├──────────────────────────────────────┼─────────────────────────────┼
─────────────────┤
│ activation_10 (Activation)           │ (None, 7)                   │
0 │
└──────────────────────────────────────┴─────────────────────────────┴
─────────────────┘

 Total params: 1,328,167 (5.07 MB)

 Trainable params: 1,325,991 (5.06 MB)

 Non-trainable params: 2,176 (8.50 KB)

None
```

```python
from keras.optimizers import RMSprop,SGD,Adam
from keras.callbacks import ModelCheckpoint, EarlyStopping,
ReduceLROnPlateau

checkpoint = ModelCheckpoint(r'C:\em3\Emotion_little_vgg.keras',
                             monitor='val_loss',
                             mode='min',
                             save_best_only=True,
                             verbose=1)
```

```
earlystop = EarlyStopping(monitor='val_loss',
                          min_delta=0,
                          patience=5,
                          verbose=1,
                          restore_best_weights=True
                          )

reduce_lr = ReduceLROnPlateau(monitor='val_loss',
                              factor=0.2,
                              patience=3,
                              verbose=1,
                              min_delta=0.0001)

callbacks = [earlystop,checkpoint,reduce_lr]

model.compile(loss='categorical_crossentropy',
              optimizer = Adam(learning_rate=0.001),
              metrics=['accuracy'])

import os

# Function to count valid images in a directory
def count_valid_images_in_directory(directory):
    count = 0
    for root, dirs, files in os.walk(directory):
        for file in files:
            if file.endswith(('.png', '.jpg', '.jpeg')):
                count += 1
    return count

nb_train_samples = count_valid_images_in_directory(train_data_dir)
nb_validation_samples =
count_valid_images_in_directory(validation_data_dir)
print(nb_train_samples)
print(nb_validation_samples)

28709
7178

epochs=75

history=model.fit(
                train_generator,
                steps_per_epoch=nb_train_samples//batch_size,
                epochs=epochs,
                callbacks=callbacks,
                validation_data=validation_generator,
                validation_steps=nb_validation_samples//batch_size)

Epoch 1/75
```

```
C:\Users\Asus\AppData\Roaming\Python\Python312\site-packages\keras\
src\trainers\data_adapters\py_dataset_adapter.py:121: UserWarning:
Your `PyDataset` class should call `super().__init__(**kwargs)` in its
constructor. `**kwargs` can include `workers`, `use_multiprocessing`,
`max_queue_size`. Do not pass these arguments to `fit()`, as they will
be ignored.
  self._warn_if_super_not_called()

897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 102ms/step - accuracy: 0.1751 - loss:
2.4100

C:\Users\Asus\AppData\Roaming\Python\Python312\site-packages\keras\
src\trainers\data_adapters\py_dataset_adapter.py:121: UserWarning:
Your `PyDataset` class should call `super().__init__(**kwargs)` in its
constructor. `**kwargs` can include `workers`, `use_multiprocessing`,
`max_queue_size`. Do not pass these arguments to `fit()`, as they will
be ignored.
  self._warn_if_super_not_called()


Epoch 1: val_loss improved from inf to 1.77859, saving model to C:\
em3\Emotion_little_vgg.keras
897/897 ━━━━━━━━━━━━━━━━━━━━ 102s 110ms/step - accuracy: 0.1751 -
loss: 2.4097 - val_accuracy: 0.2564 - val_loss: 1.7786 -
learning_rate: 0.0010
Epoch 2/75
  1/897 ━━━━━━━━━━━━━━━━━━━━ 2:05 140ms/step - accuracy: 0.3750 -
loss: 1.6353
Epoch 2: val_loss improved from 1.77859 to 1.74256, saving model to
C:\em3\Emotion_little_vgg.keras
897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 202us/step - accuracy: 0.3750 - loss:
1.6353 - val_accuracy: 0.3000 - val_loss: 1.7426 - learning_rate:
0.0010
Epoch 3/75

C:\Users\Asus\anaconda3\Lib\contextlib.py:158: UserWarning: Your input
ran out of data; interrupting training. Make sure that your dataset or
generator can generate at least `steps_per_epoch * epochs` batches.
You may need to use the `.repeat()` function when building your
dataset.
  self.gen.throw(value)

897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.2351 - loss:
1.8274
Epoch 3: val_loss did not improve from 1.74256
897/897 ━━━━━━━━━━━━━━━━━━━━ 111s 124ms/step - accuracy: 0.2351 -
loss: 1.8274 - val_accuracy: 0.2607 - val_loss: 1.7685 -
learning_rate: 0.0010
Epoch 4/75
  1/897 ━━━━━━━━━━━━━━━━━━━━ 1:39 111ms/step - accuracy: 0.1562 -
```

```
loss: 1.9854
Epoch 4: val_loss improved from 1.74256 to 1.64791, saving model to
C:\em3\Emotion_little_vgg.keras
897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 165us/step - accuracy: 0.1562 - loss:
1.9854 - val_accuracy: 0.2000 - val_loss: 1.6479 - learning_rate:
0.0010
Epoch 5/75
897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - accuracy: 0.2510 - loss:
1.7992
Epoch 5: val_loss did not improve from 1.64791
897/897 ━━━━━━━━━━━━━━━━━━━━ 116s 129ms/step - accuracy: 0.2510 -
loss: 1.7992 - val_accuracy: 0.2739 - val_loss: 1.7542 -
learning_rate: 0.0010
Epoch 6/75
  1/897 ━━━━━━━━━━━━━━━━━━━━ 1:59 133ms/step - accuracy: 0.2188 -
loss: 1.8228
Epoch 6: val_loss did not improve from 1.64791
897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 39us/step - accuracy: 0.2188 - loss:
1.8228 - val_accuracy: 0.3000 - val_loss: 1.7767 - learning_rate:
0.0010
Epoch 7/75
897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 122ms/step - accuracy: 0.2650 - loss:
1.7713
Epoch 7: val_loss improved from 1.64791 to 1.60297, saving model to
C:\em3\Emotion_little_vgg.keras
897/897 ━━━━━━━━━━━━━━━━━━━━ 117s 131ms/step - accuracy: 0.2650 -
loss: 1.7713 - val_accuracy: 0.3562 - val_loss: 1.6030 -
learning_rate: 0.0010
Epoch 8/75
  1/897 ━━━━━━━━━━━━━━━━━━━━ 2:14 150ms/step - accuracy: 0.3750 -
loss: 1.7746
Epoch 8: val_loss did not improve from 1.60297
897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 34us/step - accuracy: 0.3750 - loss:
1.7746 - val_accuracy: 0.1000 - val_loss: 1.9899 - learning_rate:
0.0010
Epoch 9/75
897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 125ms/step - accuracy: 0.3091 - loss:
1.7035
Epoch 9: val_loss did not improve from 1.60297
897/897 ━━━━━━━━━━━━━━━━━━━━ 120s 133ms/step - accuracy: 0.3091 -
loss: 1.7035 - val_accuracy: 0.4058 - val_loss: 1.6697 -
learning_rate: 0.0010
Epoch 10/75
  1/897 ━━━━━━━━━━━━━━━━━━━━ 1:46 119ms/step - accuracy: 0.2188 -
loss: 1.5502
Epoch 10: val_loss did not improve from 1.60297

Epoch 10: ReduceLROnPlateau reducing learning rate to
0.00020000000949949026.
```

```
897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 45us/step - accuracy: 0.2188 - loss:
1.5502 - val_accuracy: 0.4000 - val_loss: 1.6191 - learning_rate:
0.0010
Epoch 11/75
897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 126ms/step - accuracy: 0.3772 - loss:
1.5904
Epoch 11: val_loss improved from 1.60297 to 1.36923, saving model to
C:\em3\Emotion_little_vgg.keras
897/897 ━━━━━━━━━━━━━━━━━━━━ 121s 135ms/step - accuracy: 0.3772 -
loss: 1.5904 - val_accuracy: 0.4795 - val_loss: 1.3692 -
learning_rate: 2.0000e-04
Epoch 12/75
  1/897 ━━━━━━━━━━━━━━━━━━━━ 1:57 131ms/step - accuracy: 0.5312 -
loss: 1.2942
Epoch 12: val_loss did not improve from 1.36923
897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 37us/step - accuracy: 0.5312 - loss:
1.2942 - val_accuracy: 0.4000 - val_loss: 1.5575 - learning_rate:
2.0000e-04
Epoch 13/75
897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 128ms/step - accuracy: 0.3912 - loss:
1.5547
Epoch 13: val_loss improved from 1.36923 to 1.33923, saving model to
C:\em3\Emotion_little_vgg.keras
897/897 ━━━━━━━━━━━━━━━━━━━━ 123s 137ms/step - accuracy: 0.3912 -
loss: 1.5547 - val_accuracy: 0.4929 - val_loss: 1.3392 -
learning_rate: 2.0000e-04
Epoch 14/75
  1/897 ━━━━━━━━━━━━━━━━━━━━ 2:05 140ms/step - accuracy: 0.4688 -
loss: 1.4764
Epoch 14: val_loss did not improve from 1.33923
897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 42us/step - accuracy: 0.4688 - loss:
1.4764 - val_accuracy: 0.4000 - val_loss: 1.5781 - learning_rate:
2.0000e-04
Epoch 15/75
897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 130ms/step - accuracy: 0.4114 - loss:
1.5131
Epoch 15: val_loss improved from 1.33923 to 1.29221, saving model to
C:\em3\Emotion_little_vgg.keras
897/897 ━━━━━━━━━━━━━━━━━━━━ 125s 139ms/step - accuracy: 0.4114 -
loss: 1.5131 - val_accuracy: 0.5080 - val_loss: 1.2922 -
learning_rate: 2.0000e-04
Epoch 16/75
  1/897 ━━━━━━━━━━━━━━━━━━━━ 2:00 135ms/step - accuracy: 0.4062 -
loss: 1.6187
Epoch 16: val_loss did not improve from 1.29221
897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 45us/step - accuracy: 0.4062 - loss:
1.6187 - val_accuracy: 0.3000 - val_loss: 1.4279 - learning_rate:
2.0000e-04
Epoch 17/75
```

```
897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 131ms/step - accuracy: 0.4264 - loss:
1.4974
Epoch 17: val_loss improved from 1.29221 to 1.27873, saving model to
C:\em3\Emotion_little_vgg.keras
897/897 ━━━━━━━━━━━━━━━━━━━━ 125s 140ms/step - accuracy: 0.4265 -
loss: 1.4974 - val_accuracy: 0.5117 - val_loss: 1.2787 -
learning_rate: 2.0000e-04
Epoch 18/75
  1/897 ━━━━━━━━━━━━━━━━━━━━ 2:46 185ms/step - accuracy: 0.4375 -
loss: 1.4669
Epoch 18: val_loss improved from 1.27873 to 0.87588, saving model to
C:\em3\Emotion_little_vgg.keras
897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 201us/step - accuracy: 0.4375 - loss:
1.4669 - val_accuracy: 0.7000 - val_loss: 0.8759 - learning_rate:
2.0000e-04
Epoch 19/75
897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 132ms/step - accuracy: 0.4255 - loss:
1.4761
Epoch 19: val_loss did not improve from 0.87588
897/897 ━━━━━━━━━━━━━━━━━━━━ 126s 141ms/step - accuracy: 0.4255 -
loss: 1.4761 - val_accuracy: 0.5148 - val_loss: 1.2523 -
learning_rate: 2.0000e-04
Epoch 20/75
  1/897 ━━━━━━━━━━━━━━━━━━━━ 1:57 132ms/step - accuracy: 0.4688 -
loss: 1.4711
Epoch 20: val_loss did not improve from 0.87588
897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 40us/step - accuracy: 0.4688 - loss:
1.4711 - val_accuracy: 0.8000 - val_loss: 0.9409 - learning_rate:
2.0000e-04
Epoch 21/75
897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 130ms/step - accuracy: 0.4480 - loss:
1.4403
Epoch 21: val_loss did not improve from 0.87588

Epoch 21: ReduceLROnPlateau reducing learning rate to
4.0000001899898055e-05.
897/897 ━━━━━━━━━━━━━━━━━━━━ 125s 139ms/step - accuracy: 0.4480 -
loss: 1.4403 - val_accuracy: 0.5084 - val_loss: 1.2503 -
learning_rate: 2.0000e-04
Epoch 22/75
  1/897 ━━━━━━━━━━━━━━━━━━━━ 1:49 122ms/step - accuracy: 0.5312 -
loss: 1.3272
Epoch 22: val_loss did not improve from 0.87588
897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 37us/step - accuracy: 0.5312 - loss:
1.3272 - val_accuracy: 0.2000 - val_loss: 1.7289 - learning_rate:
4.0000e-05
Epoch 23/75
897/897 ━━━━━━━━━━━━━━━━━━━━ 0s 132ms/step - accuracy: 0.4476 - loss:
1.4345
```

```
Epoch 23: val_loss did not improve from 0.87588
897/897 ━━━━━━━━━━━━━━━━━━━━ 126s 141ms/step - accuracy: 0.4476 -
loss: 1.4345 - val_accuracy: 0.5306 - val_loss: 1.2021 -
learning_rate: 4.0000e-05
Epoch 23: early stopping
Restoring model weights from the end of the best epoch: 18.
```

```
model.save(r'C:\em3\Emotion_little_final_vgg.keras')
```