mage-classification-using-resnet50

October 27, 2024

[ ]:
```
import os
from shutil import copy
from collections import defaultdict
from google.colab import drive
drive.mount('/content/drive')

extract_to_360_path = '/content/drive/My Drive/360 Rocks'
extract_to_120_path = '/content/drive/My Drive/120 Rocks'
```

```
Mounted at /content/drive
```

1. From the '360 rocks' folder create a training dataset that has 30 categories and 12 images in each category. To create the categories, group the rocks by name: e.g., one category will be 12 Andesite rocks, another category will be 12 Basalt rocks, etc. From '120 rocks' folder create a validation dataset that has 30 categories and 4 images in each category.

[ ]:
```
import shutil
import os

def create_datasets(source_folder, dest_folder, num_categories,␣
 ↪num_images_per_category, numItemsToExclude):
    categories = defaultdict(list)
    for filename in os.listdir(source_folder):
        if filename.endswith('.jpg'):
            category_name = '_'.join(filename.split('_')[:-numItemsToExclude])
            categories[category_name].append(filename)
    if not os.path.exists(dest_folder):
        os.makedirs(dest_folder)

    for category, files in categories.items():
        if len(files) >= num_images_per_category:
            category_folder = os.path.join(dest_folder, category)
            os.makedirs(category_folder, exist_ok=True)
            for file in files[:num_images_per_category]:
                source_path = os.path.join(source_folder, file)
                dest_path = os.path.join(category_folder, file)
                copy(source_path, dest_path)
```

```python
train_folder = '/content/drive/My Drive/training_dataset'
val_folder = '/content/drive/My Drive/validation_dataset'

if os.path.exists(train_folder):
    shutil.rmtree(train_folder)

if os.path.exists(val_folder):
    shutil.rmtree(val_folder)


create_datasets(extract_to_360_path, train_folder, 30, 12, 1)
create_datasets(extract_to_120_path, val_folder, 30, 4, 2)
```

2. Build the input pipeline, including the appropriate preprocessing operations, and add data augmentation.

[ ]:
```python
import os
import numpy as np
import tensorflow as tf
import random
from tensorflow.keras.preprocessing.image import ImageDataGenerator

random.seed(42)
np.random.seed(42)
tf.random.set_seed(42)

train_dir = '/content/drive/My Drive/training_dataset'
val_dir = '/content/drive/My Drive/validation_dataset'

train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

val_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(224, 224),
    batch_size=16,
```

```
    class_mode='categorical'
)

val_generator = val_datagen.flow_from_directory(
    val_dir,
    target_size=(224, 224),
    batch_size=16,
    class_mode='categorical'
)
```

```
Found 360 images belonging to 30 classes.
Found 120 images belonging to 30 classes.
```

3. Fine-tune a pretrained neural network of your choice on the training data from the previous step. Before fine-tuning, you should remove the top layer of the pretrained network and add at least two more layers with a softmax activation function (your output layer should have 30 neurons and the layer before the output layer should have 8 neurons). For fine-tuning, you should first train for a few epochs only weights in the layers, and after that, you should train all the weights (you should try to run your code until convergence, but for at least 20 epochs). Display train and validation loss and accuracy during the training and make a plot showing train and validation loss and accuracy as a function of the training epoch (mark the epoch where you switched to training the whole network with a vertical line).

[ ]:
```
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, Input,␣
  ↪BatchNormalization, Dropout
from tensorflow.keras.models import Model
import tensorflow as tf
from tensorflow.keras.callbacks import EarlyStopping
import json

base_model = ResNet50(weights='imagenet', include_top=False,␣
  ↪input_tensor=Input(shape=(224, 224, 3)))

for layer in base_model.layers:
    layer.trainable = False

x = GlobalAveragePooling2D()(base_model.output)
x = Dense(512, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.
  ↪01))(x)
x = BatchNormalization()(x)
x = Dropout(0.5)(x)
x = Dense(128, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.
  ↪01))(x)
x = BatchNormalization()(x)
x = Dropout(0.25)(x)
```

```
x = Dense(8, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.
 ↪01))(x)
predictions = Dense(30, activation='softmax')(x)
model = Model(inputs=base_model.input, outputs=predictions)

opt = tf.keras.optimizers.Adam(learning_rate=1e-4)

model.compile(optimizer=opt, loss='categorical_crossentropy',␣
 ↪metrics=['accuracy'])
early_stopping = EarlyStopping(monitor='val_loss', patience=3,␣
 ↪restore_best_weights=True)


history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.n // train_generator.batch_size,
    epochs=5,
    validation_data=val_generator,
    validation_steps=val_generator.n // val_generator.batch_size
)

for layer in base_model.layers:
    layer.trainable = True

opt = tf.keras.optimizers.Adam(learning_rate=1e-5)

model.compile(optimizer=opt, loss='categorical_crossentropy',␣
 ↪metrics=['accuracy'])

fine_tune_epochs = 245
total_epochs = 5 + fine_tune_epochs

history_fine = model.fit(
    train_generator,
    steps_per_epoch=train_generator.n // train_generator.batch_size,
    epochs=total_epochs,
    initial_epoch=history.epoch[-1],
    validation_data=val_generator,
    validation_steps=val_generator.n // val_generator.batch_size,
    callbacks=[early_stopping]
)

model_save_path = '/content/drive/My Drive/my_model.h5'
model.save(model_save_path)

history_save_path = '/content/drive/My Drive/my_history.json'
with open(history_save_path, 'w') as f:
```

```
    json.dump(history.history, f)
```

```
Epoch 1/5
22/22 [==============================] - 15s 391ms/step - loss: 3.5267 -
accuracy: 0.0465 - val_loss: 3.4114 - val_accuracy: 0.0179
Epoch 2/5
22/22 [==============================] - 13s 601ms/step - loss: 3.4765 -
accuracy: 0.0465 - val_loss: 3.4063 - val_accuracy: 0.0268
Epoch 3/5
22/22 [==============================] - 9s 409ms/step - loss: 3.4954 -
accuracy: 0.0465 - val_loss: 3.4108 - val_accuracy: 0.0357
Epoch 4/5
22/22 [==============================] - 9s 421ms/step - loss: 3.4516 -
accuracy: 0.0552 - val_loss: 3.4287 - val_accuracy: 0.0446
Epoch 5/5
22/22 [==============================] - 10s 472ms/step - loss: 3.4421 -
accuracy: 0.0640 - val_loss: 3.4378 - val_accuracy: 0.0357
Epoch 5/250
22/22 [==============================] - 48s 482ms/step - loss: 3.5209 -
accuracy: 0.0291 - val_loss: 3.4367 - val_accuracy: 0.0446
Epoch 6/250
22/22 [==============================] - 10s 423ms/step - loss: 3.5607 -
accuracy: 0.0291 - val_loss: 3.4751 - val_accuracy: 0.0357
Epoch 7/250
22/22 [==============================] - 14s 638ms/step - loss: 3.5221 -
accuracy: 0.0378 - val_loss: 3.4891 - val_accuracy: 0.0357
Epoch 8/250
22/22 [==============================] - 12s 551ms/step - loss: 3.4782 -
accuracy: 0.0378 - val_loss: 3.5394 - val_accuracy: 0.0268
Epoch 9/250
22/22 [==============================] - 10s 457ms/step - loss: 3.4497 -
accuracy: 0.0465 - val_loss: 3.5725 - val_accuracy: 0.0357
Epoch 10/250
22/22 [==============================] - 10s 466ms/step - loss: 3.4133 -
accuracy: 0.0436 - val_loss: 3.6825 - val_accuracy: 0.0446
Epoch 11/250
22/22 [==============================] - 9s 388ms/step - loss: 3.4098 -
accuracy: 0.0407 - val_loss: 3.7174 - val_accuracy: 0.0357
Epoch 12/250
22/22 [==============================] - 11s 501ms/step - loss: 3.3776 -
accuracy: 0.0640 - val_loss: 3.8610 - val_accuracy: 0.0357
Epoch 13/250
22/22 [==============================] - 10s 471ms/step - loss: 3.3479 -
accuracy: 0.0610 - val_loss: 3.8687 - val_accuracy: 0.0357
Epoch 14/250
22/22 [==============================] - 9s 383ms/step - loss: 3.2628 -
accuracy: 0.0814 - val_loss: 3.9045 - val_accuracy: 0.0268
```

```
Epoch 15/250
22/22 [==============================] - 14s 637ms/step - loss: 3.2812 -
accuracy: 0.0930 - val_loss: 3.9658 - val_accuracy: 0.0179
Epoch 16/250
22/22 [==============================] - 12s 552ms/step - loss: 3.2297 -
accuracy: 0.0901 - val_loss: 4.0320 - val_accuracy: 0.0179
Epoch 17/250
22/22 [==============================] - 10s 442ms/step - loss: 3.1890 -
accuracy: 0.1192 - val_loss: 4.0927 - val_accuracy: 0.0089
Epoch 18/250
22/22 [==============================] - 8s 380ms/step - loss: 3.2104 -
accuracy: 0.1134 - val_loss: 4.1212 - val_accuracy: 0.0268
Epoch 19/250
22/22 [==============================] - 13s 604ms/step - loss: 3.1358 -
accuracy: 0.1105 - val_loss: 4.2273 - val_accuracy: 0.0179
Epoch 20/250
22/22 [==============================] - 11s 508ms/step - loss: 3.1708 -
accuracy: 0.0988 - val_loss: 4.2243 - val_accuracy: 0.0089
Epoch 21/250
22/22 [==============================] - 12s 570ms/step - loss: 3.0866 -
accuracy: 0.1163 - val_loss: 4.1612 - val_accuracy: 0.0179
Epoch 22/250
22/22 [==============================] - 12s 526ms/step - loss: 3.0609 -
accuracy: 0.1483 - val_loss: 4.1582 - val_accuracy: 0.0179
Epoch 23/250
22/22 [==============================] - 11s 501ms/step - loss: 3.0482 -
accuracy: 0.1424 - val_loss: 4.2023 - val_accuracy: 0.0179
Epoch 24/250
22/22 [==============================] - 10s 440ms/step - loss: 3.0460 -
accuracy: 0.1424 - val_loss: 4.0887 - val_accuracy: 0.0268
Epoch 25/250
22/22 [==============================] - 9s 386ms/step - loss: 3.0242 -
accuracy: 0.1512 - val_loss: 4.1729 - val_accuracy: 0.0179
Epoch 26/250
22/22 [==============================] - 10s 432ms/step - loss: 2.9748 -
accuracy: 0.1715 - val_loss: 4.1646 - val_accuracy: 0.0089
Epoch 27/250
22/22 [==============================] - 11s 502ms/step - loss: 2.9261 -
accuracy: 0.1512 - val_loss: 4.1208 - val_accuracy: 0.0179
Epoch 28/250
22/22 [==============================] - 10s 467ms/step - loss: 2.9303 -
accuracy: 0.1657 - val_loss: 4.0283 - val_accuracy: 0.0357
Epoch 29/250
22/22 [==============================] - 8s 380ms/step - loss: 2.9352 -
accuracy: 0.1453 - val_loss: 4.1026 - val_accuracy: 0.0357
Epoch 30/250
22/22 [==============================] - 9s 383ms/step - loss: 2.8756 -
accuracy: 0.1715 - val_loss: 4.0430 - val_accuracy: 0.0446
```

```
Epoch 31/250
22/22 [==============================] - 11s 497ms/step - loss: 2.9354 -
accuracy: 0.1860 - val_loss: 3.9924 - val_accuracy: 0.0357
Epoch 32/250
22/22 [==============================] - 10s 442ms/step - loss: 2.8635 -
accuracy: 0.1453 - val_loss: 4.0712 - val_accuracy: 0.0268
Epoch 33/250
22/22 [==============================] - 10s 463ms/step - loss: 2.8727 -
accuracy: 0.1773 - val_loss: 4.0577 - val_accuracy: 0.0268
Epoch 34/250
22/22 [==============================] - 9s 382ms/step - loss: 2.8243 -
accuracy: 0.1831 - val_loss: 3.9722 - val_accuracy: 0.0357
Epoch 35/250
22/22 [==============================] - 9s 407ms/step - loss: 2.8602 -
accuracy: 0.1860 - val_loss: 4.0291 - val_accuracy: 0.0268
Epoch 36/250
22/22 [==============================] - 11s 494ms/step - loss: 2.8213 -
accuracy: 0.1686 - val_loss: 3.9542 - val_accuracy: 0.0446
Epoch 37/250
22/22 [==============================] - 10s 439ms/step - loss: 2.8199 -
accuracy: 0.1890 - val_loss: 3.9505 - val_accuracy: 0.0357
Epoch 38/250
22/22 [==============================] - 10s 445ms/step - loss: 2.7502 -
accuracy: 0.2151 - val_loss: 3.9369 - val_accuracy: 0.0357
Epoch 39/250
22/22 [==============================] - 10s 443ms/step - loss: 2.7972 -
accuracy: 0.2035 - val_loss: 3.8576 - val_accuracy: 0.0625
Epoch 40/250
22/22 [==============================] - 11s 494ms/step - loss: 2.6861 -
accuracy: 0.2267 - val_loss: 3.8625 - val_accuracy: 0.0446
Epoch 41/250
22/22 [==============================] - 11s 500ms/step - loss: 2.7212 -
accuracy: 0.2122 - val_loss: 3.8079 - val_accuracy: 0.0536
Epoch 42/250
22/22 [==============================] - 10s 445ms/step - loss: 2.7235 -
accuracy: 0.2122 - val_loss: 3.7355 - val_accuracy: 0.0446
Epoch 43/250
22/22 [==============================] - 9s 398ms/step - loss: 2.6525 -
accuracy: 0.2267 - val_loss: 3.6635 - val_accuracy: 0.0357
Epoch 44/250
22/22 [==============================] - 11s 503ms/step - loss: 2.6823 -
accuracy: 0.2151 - val_loss: 3.6051 - val_accuracy: 0.0446
Epoch 45/250
22/22 [==============================] - 11s 499ms/step - loss: 2.6677 -
accuracy: 0.2238 - val_loss: 3.5181 - val_accuracy: 0.0714
Epoch 46/250
22/22 [==============================] - 11s 507ms/step - loss: 2.6376 -
accuracy: 0.2471 - val_loss: 3.4914 - val_accuracy: 0.0625
```

```
Epoch 47/250
22/22 [==============================] - 10s 460ms/step - loss: 2.6048 -
accuracy: 0.2674 - val_loss: 3.4220 - val_accuracy: 0.0714
Epoch 48/250
22/22 [==============================] - 10s 437ms/step - loss: 2.5816 -
accuracy: 0.2587 - val_loss: 3.3673 - val_accuracy: 0.0804
Epoch 49/250
22/22 [==============================] - 11s 472ms/step - loss: 2.5557 -
accuracy: 0.2616 - val_loss: 3.3520 - val_accuracy: 0.0714
Epoch 50/250
22/22 [==============================] - 11s 495ms/step - loss: 2.5717 -
accuracy: 0.2587 - val_loss: 3.3322 - val_accuracy: 0.0714
Epoch 51/250
22/22 [==============================] - 10s 445ms/step - loss: 2.5363 -
accuracy: 0.2727 - val_loss: 3.2965 - val_accuracy: 0.0893
Epoch 52/250
22/22 [==============================] - 10s 439ms/step - loss: 2.5100 -
accuracy: 0.2733 - val_loss: 3.2338 - val_accuracy: 0.1161
Epoch 53/250
22/22 [==============================] - 9s 385ms/step - loss: 2.5425 -
accuracy: 0.2703 - val_loss: 3.1444 - val_accuracy: 0.1250
Epoch 54/250
22/22 [==============================] - 10s 435ms/step - loss: 2.4746 -
accuracy: 0.2703 - val_loss: 3.1600 - val_accuracy: 0.1071
Epoch 55/250
22/22 [==============================] - 10s 445ms/step - loss: 2.4397 -
accuracy: 0.2994 - val_loss: 3.1231 - val_accuracy: 0.1071
Epoch 56/250
22/22 [==============================] - 10s 445ms/step - loss: 2.4851 -
accuracy: 0.2733 - val_loss: 3.1586 - val_accuracy: 0.0804
Epoch 57/250
22/22 [==============================] - 9s 382ms/step - loss: 2.4883 -
accuracy: 0.2936 - val_loss: 3.0994 - val_accuracy: 0.0982
Epoch 58/250
22/22 [==============================] - 11s 494ms/step - loss: 2.5139 -
accuracy: 0.2849 - val_loss: 3.1313 - val_accuracy: 0.0893
Epoch 59/250
22/22 [==============================] - 11s 504ms/step - loss: 2.4268 -
accuracy: 0.3081 - val_loss: 3.1042 - val_accuracy: 0.0982
Epoch 60/250
22/22 [==============================] - 10s 442ms/step - loss: 2.4734 -
accuracy: 0.2878 - val_loss: 3.0785 - val_accuracy: 0.0804
Epoch 61/250
22/22 [==============================] - 10s 448ms/step - loss: 2.4786 -
accuracy: 0.2703 - val_loss: 3.0063 - val_accuracy: 0.1250
Epoch 62/250
22/22 [==============================] - 11s 511ms/step - loss: 2.3946 -
accuracy: 0.3052 - val_loss: 3.0601 - val_accuracy: 0.1250
```

```
Epoch 63/250
22/22 [==============================] - 11s 503ms/step - loss: 2.4353 -
accuracy: 0.2994 - val_loss: 3.1015 - val_accuracy: 0.1071
Epoch 64/250
22/22 [==============================] - 10s 448ms/step - loss: 2.3602 -
accuracy: 0.3110 - val_loss: 3.0581 - val_accuracy: 0.1339
Epoch 65/250
22/22 [==============================] - 9s 392ms/step - loss: 2.3453 -
accuracy: 0.3227 - val_loss: 3.0988 - val_accuracy: 0.1518
Epoch 66/250
22/22 [==============================] - 9s 407ms/step - loss: 2.3708 -
accuracy: 0.3081 - val_loss: 3.0964 - val_accuracy: 0.1607
Epoch 67/250
22/22 [==============================] - 11s 524ms/step - loss: 2.3449 -
accuracy: 0.3466 - val_loss: 3.0599 - val_accuracy: 0.1696
Epoch 68/250
22/22 [==============================] - 11s 514ms/step - loss: 2.3367 -
accuracy: 0.3314 - val_loss: 3.0587 - val_accuracy: 0.1429
Epoch 69/250
22/22 [==============================] - 10s 444ms/step - loss: 2.3674 -
accuracy: 0.3256 - val_loss: 3.0690 - val_accuracy: 0.1607
Epoch 70/250
22/22 [==============================] - 9s 398ms/step - loss: 2.3051 -
accuracy: 0.3198 - val_loss: 3.1121 - val_accuracy: 0.1429
Epoch 71/250
22/22 [==============================] - 10s 455ms/step - loss: 2.2707 -
accuracy: 0.3576 - val_loss: 3.0479 - val_accuracy: 0.1696
Epoch 72/250
22/22 [==============================] - 11s 502ms/step - loss: 2.2912 -
accuracy: 0.3140 - val_loss: 3.0530 - val_accuracy: 0.1607
Epoch 73/250
22/22 [==============================] - 10s 442ms/step - loss: 2.2735 -
accuracy: 0.3285 - val_loss: 3.0606 - val_accuracy: 0.1339
Epoch 74/250
22/22 [==============================] - 9s 379ms/step - loss: 2.2940 -
accuracy: 0.3547 - val_loss: 3.1054 - val_accuracy: 0.1429
Epoch 75/250
22/22 [==============================] - 10s 441ms/step - loss: 2.2369 -
accuracy: 0.3750 - val_loss: 3.0666 - val_accuracy: 0.1696
Epoch 76/250
22/22 [==============================] - 11s 487ms/step - loss: 2.2115 -
accuracy: 0.3837 - val_loss: 3.0779 - val_accuracy: 0.1696
Epoch 77/250
22/22 [==============================] - 10s 442ms/step - loss: 2.2041 -
accuracy: 0.3953 - val_loss: 3.0472 - val_accuracy: 0.1875
Epoch 78/250
22/22 [==============================] - 11s 495ms/step - loss: 2.2307 -
accuracy: 0.3779 - val_loss: 3.1065 - val_accuracy: 0.1607
```

```
Epoch 79/250
22/22 [==============================] - 11s 502ms/step - loss: 2.1812 -
accuracy: 0.3895 - val_loss: 3.0265 - val_accuracy: 0.1786
Epoch 80/250
22/22 [==============================] - 9s 402ms/step - loss: 2.1475 -
accuracy: 0.4157 - val_loss: 3.0450 - val_accuracy: 0.1696
Epoch 81/250
22/22 [==============================] - 10s 458ms/step - loss: 2.1506 -
accuracy: 0.3693 - val_loss: 3.0975 - val_accuracy: 0.1607
Epoch 82/250
22/22 [==============================] - 11s 485ms/step - loss: 2.1763 -
accuracy: 0.3634 - val_loss: 3.0850 - val_accuracy: 0.1696
Epoch 83/250
22/22 [==============================] - 10s 455ms/step - loss: 2.1885 -
accuracy: 0.3808 - val_loss: 3.0768 - val_accuracy: 0.1875
Epoch 84/250
22/22 [==============================] - 11s 492ms/step - loss: 2.1082 -
accuracy: 0.3692 - val_loss: 3.0795 - val_accuracy: 0.1964
Epoch 85/250
22/22 [==============================] - 9s 396ms/step - loss: 2.1050 -
accuracy: 0.3983 - val_loss: 3.0330 - val_accuracy: 0.2054
Epoch 86/250
22/22 [==============================] - 11s 520ms/step - loss: 2.1173 -
accuracy: 0.4099 - val_loss: 2.9899 - val_accuracy: 0.2143
Epoch 87/250
22/22 [==============================] - 11s 505ms/step - loss: 2.0716 -
accuracy: 0.4273 - val_loss: 2.9542 - val_accuracy: 0.2143
Epoch 88/250
22/22 [==============================] - 10s 452ms/step - loss: 2.0808 -
accuracy: 0.4419 - val_loss: 2.9827 - val_accuracy: 0.2232
Epoch 89/250
22/22 [==============================] - 10s 459ms/step - loss: 2.0935 -
accuracy: 0.4273 - val_loss: 3.0207 - val_accuracy: 0.1786
Epoch 90/250
22/22 [==============================] - 10s 417ms/step - loss: 2.0313 -
accuracy: 0.4331 - val_loss: 2.9899 - val_accuracy: 0.1696
Epoch 91/250
22/22 [==============================] - 10s 468ms/step - loss: 2.0346 -
accuracy: 0.4419 - val_loss: 2.9998 - val_accuracy: 0.1964
Epoch 92/250
22/22 [==============================] - 10s 441ms/step - loss: 2.0570 -
accuracy: 0.4273 - val_loss: 3.0436 - val_accuracy: 0.1786
Epoch 93/250
22/22 [==============================] - 10s 448ms/step - loss: 2.0422 -
accuracy: 0.4448 - val_loss: 2.9931 - val_accuracy: 0.1964
Epoch 94/250
22/22 [==============================] - 10s 446ms/step - loss: 2.0147 -
accuracy: 0.4448 - val_loss: 2.9949 - val_accuracy: 0.1964
```

```
Epoch 95/250
22/22 [==============================] - 11s 507ms/step - loss: 2.0151 -
accuracy: 0.4506 - val_loss: 2.9465 - val_accuracy: 0.1875
Epoch 96/250
22/22 [==============================] - 10s 471ms/step - loss: 1.9961 -
accuracy: 0.4535 - val_loss: 2.9867 - val_accuracy: 0.1875
Epoch 97/250
22/22 [==============================] - 9s 406ms/step - loss: 2.0342 -
accuracy: 0.4215 - val_loss: 2.9576 - val_accuracy: 0.1786
Epoch 98/250
22/22 [==============================] - 10s 465ms/step - loss: 1.9863 -
accuracy: 0.4535 - val_loss: 3.0120 - val_accuracy: 0.1786
Epoch 99/250
22/22 [==============================] - 10s 464ms/step - loss: 1.9510 -
accuracy: 0.4535 - val_loss: 2.9510 - val_accuracy: 0.1786
Epoch 100/250
22/22 [==============================] - 10s 447ms/step - loss: 2.0142 -
accuracy: 0.4070 - val_loss: 2.9749 - val_accuracy: 0.1875
Epoch 101/250
22/22 [==============================] - 9s 408ms/step - loss: 1.9628 -
accuracy: 0.4302 - val_loss: 2.9888 - val_accuracy: 0.2143
Epoch 102/250
22/22 [==============================] - 10s 477ms/step - loss: 1.8906 -
accuracy: 0.4971 - val_loss: 2.9340 - val_accuracy: 0.1875
Epoch 103/250
22/22 [==============================] - 11s 480ms/step - loss: 1.9289 -
accuracy: 0.4535 - val_loss: 2.9673 - val_accuracy: 0.1696
Epoch 104/250
22/22 [==============================] - 11s 477ms/step - loss: 1.8928 -
accuracy: 0.4506 - val_loss: 2.9557 - val_accuracy: 0.2143
Epoch 105/250
22/22 [==============================] - 11s 511ms/step - loss: 1.9641 -
accuracy: 0.4709 - val_loss: 2.9823 - val_accuracy: 0.1964
Epoch 106/250
22/22 [==============================] - 11s 519ms/step - loss: 1.8603 -
accuracy: 0.5087 - val_loss: 3.0098 - val_accuracy: 0.1875
Epoch 107/250
22/22 [==============================] - 12s 534ms/step - loss: 1.9104 -
accuracy: 0.4535 - val_loss: 3.0127 - val_accuracy: 0.1875
Epoch 108/250
22/22 [==============================] - 10s 468ms/step - loss: 1.8577 -
accuracy: 0.5000 - val_loss: 2.9807 - val_accuracy: 0.1964
Epoch 109/250
22/22 [==============================] - 11s 504ms/step - loss: 1.8980 -
accuracy: 0.4767 - val_loss: 2.9737 - val_accuracy: 0.2054
Epoch 110/250
22/22 [==============================] - 12s 529ms/step - loss: 1.9273 -
accuracy: 0.4622 - val_loss: 3.0129 - val_accuracy: 0.1875
```

```
Epoch 111/250
22/22 [==============================] - 12s 535ms/step - loss: 1.8973 -
accuracy: 0.4855 - val_loss: 2.9685 - val_accuracy: 0.1964
Epoch 112/250
22/22 [==============================] - 11s 478ms/step - loss: 1.8436 -
accuracy: 0.5058 - val_loss: 3.0094 - val_accuracy: 0.1875
Epoch 113/250
22/22 [==============================] - 10s 443ms/step - loss: 1.9420 -
accuracy: 0.4767 - val_loss: 3.0052 - val_accuracy: 0.2054
Epoch 114/250
22/22 [==============================] - 9s 417ms/step - loss: 1.8470 -
accuracy: 0.4797 - val_loss: 2.9633 - val_accuracy: 0.1964
Epoch 115/250
22/22 [==============================] - 11s 517ms/step - loss: 1.8862 -
accuracy: 0.4855 - val_loss: 2.9861 - val_accuracy: 0.1429
Epoch 116/250
22/22 [==============================] - 10s 475ms/step - loss: 1.8850 -
accuracy: 0.4855 - val_loss: 3.0097 - val_accuracy: 0.1964
Epoch 117/250
22/22 [==============================] - 10s 452ms/step - loss: 1.8345 -
accuracy: 0.5291 - val_loss: 2.9624 - val_accuracy: 0.1607
Epoch 118/250
22/22 [==============================] - 9s 403ms/step - loss: 1.8061 -
accuracy: 0.5058 - val_loss: 2.9864 - val_accuracy: 0.1964
Epoch 119/250
22/22 [==============================] - 11s 524ms/step - loss: 1.7998 -
accuracy: 0.5233 - val_loss: 3.0487 - val_accuracy: 0.1786
Epoch 120/250
22/22 [==============================] - 10s 467ms/step - loss: 1.7767 -
accuracy: 0.5320 - val_loss: 3.0244 - val_accuracy: 0.1964
Epoch 121/250
22/22 [==============================] - 9s 410ms/step - loss: 1.7639 -
accuracy: 0.4884 - val_loss: 3.0368 - val_accuracy: 0.1875
Epoch 122/250
22/22 [==============================] - 11s 518ms/step - loss: 1.7232 -
accuracy: 0.5465 - val_loss: 2.9826 - val_accuracy: 0.2143
Epoch 123/250
22/22 [==============================] - 12s 527ms/step - loss: 1.7460 -
accuracy: 0.5349 - val_loss: 3.0097 - val_accuracy: 0.2321
Epoch 124/250
22/22 [==============================] - 11s 518ms/step - loss: 1.7494 -
accuracy: 0.5640 - val_loss: 2.9076 - val_accuracy: 0.2589
Epoch 125/250
22/22 [==============================] - 11s 483ms/step - loss: 1.7736 -
accuracy: 0.5262 - val_loss: 2.9932 - val_accuracy: 0.2411
Epoch 126/250
22/22 [==============================] - 10s 456ms/step - loss: 1.7144 -
accuracy: 0.5901 - val_loss: 2.9337 - val_accuracy: 0.2143
```

```
Epoch 127/250
22/22 [==============================] - 10s 420ms/step - loss: 1.7139 -
accuracy: 0.5407 - val_loss: 2.9195 - val_accuracy: 0.2054
Epoch 128/250
22/22 [==============================] - 11s 519ms/step - loss: 1.6934 -
accuracy: 0.5610 - val_loss: 3.0024 - val_accuracy: 0.2143
Epoch 129/250
22/22 [==============================] - 10s 459ms/step - loss: 1.7276 -
accuracy: 0.5349 - val_loss: 2.9812 - val_accuracy: 0.1875
Epoch 130/250
22/22 [==============================] - 10s 456ms/step - loss: 1.6842 -
accuracy: 0.5378 - val_loss: 3.0275 - val_accuracy: 0.1786
Epoch 131/250
22/22 [==============================] - 9s 405ms/step - loss: 1.6963 -
accuracy: 0.5552 - val_loss: 2.9871 - val_accuracy: 0.1964
Epoch 132/250
22/22 [==============================] - 11s 519ms/step - loss: 1.7212 -
accuracy: 0.5523 - val_loss: 2.9962 - val_accuracy: 0.2143
Epoch 133/250
22/22 [==============================] - 11s 497ms/step - loss: 1.7105 -
accuracy: 0.5465 - val_loss: 3.0093 - val_accuracy: 0.2143
Epoch 134/250
22/22 [==============================] - 10s 456ms/step - loss: 1.7020 -
accuracy: 0.5291 - val_loss: 2.9313 - val_accuracy: 0.2232
Epoch 135/250
22/22 [==============================] - 10s 465ms/step - loss: 1.6655 -
accuracy: 0.5552 - val_loss: 2.9446 - val_accuracy: 0.1964
Epoch 136/250
22/22 [==============================] - 10s 463ms/step - loss: 1.6594 -
accuracy: 0.5610 - val_loss: 2.9154 - val_accuracy: 0.2054
Epoch 137/250
22/22 [==============================] - 11s 511ms/step - loss: 1.6448 -
accuracy: 0.5901 - val_loss: 2.9251 - val_accuracy: 0.2232
Epoch 138/250
22/22 [==============================] - 11s 479ms/step - loss: 1.6583 -
accuracy: 0.5698 - val_loss: 2.9792 - val_accuracy: 0.1786
Epoch 139/250
22/22 [==============================] - 9s 400ms/step - loss: 1.6396 -
accuracy: 0.5814 - val_loss: 2.9426 - val_accuracy: 0.2143
Epoch 140/250
22/22 [==============================] - 11s 506ms/step - loss: 1.6630 -
accuracy: 0.5436 - val_loss: 2.9304 - val_accuracy: 0.2143
Epoch 141/250
22/22 [==============================] - 11s 515ms/step - loss: 1.6467 -
accuracy: 0.5436 - val_loss: 2.8647 - val_accuracy: 0.2143
Epoch 142/250
22/22 [==============================] - 11s 517ms/step - loss: 1.6039 -
accuracy: 0.5785 - val_loss: 2.9420 - val_accuracy: 0.2143
```

```
Epoch 143/250
22/22 [==============================] - 10s 473ms/step - loss: 1.6117 -
accuracy: 0.5727 - val_loss: 3.0112 - val_accuracy: 0.1964
Epoch 144/250
22/22 [==============================] - 9s 399ms/step - loss: 1.6106 -
accuracy: 0.5785 - val_loss: 2.9824 - val_accuracy: 0.2054
Epoch 145/250
22/22 [==============================] - 10s 459ms/step - loss: 1.6273 -
accuracy: 0.5872 - val_loss: 2.9536 - val_accuracy: 0.1964
Epoch 146/250
22/22 [==============================] - 11s 511ms/step - loss: 1.6225 -
accuracy: 0.5436 - val_loss: 2.9405 - val_accuracy: 0.1875
Epoch 147/250
22/22 [==============================] - 10s 467ms/step - loss: 1.5637 -
accuracy: 0.6047 - val_loss: 2.9175 - val_accuracy: 0.2054
Epoch 148/250
22/22 [==============================] - 11s 477ms/step - loss: 1.5708 -
accuracy: 0.5930 - val_loss: 2.9862 - val_accuracy: 0.2143
Epoch 149/250
22/22 [==============================] - 10s 454ms/step - loss: 1.6148 -
accuracy: 0.5872 - val_loss: 2.9731 - val_accuracy: 0.1964
Epoch 150/250
22/22 [==============================] - 10s 446ms/step - loss: 1.6155 -
accuracy: 0.5756 - val_loss: 2.9495 - val_accuracy: 0.2054
Epoch 151/250
22/22 [==============================] - 10s 447ms/step - loss: 1.5563 -
accuracy: 0.6163 - val_loss: 2.8744 - val_accuracy: 0.2232
Epoch 152/250
22/22 [==============================] - 10s 447ms/step - loss: 1.5746 -
accuracy: 0.5727 - val_loss: 2.9494 - val_accuracy: 0.2232
Epoch 153/250
22/22 [==============================] - 10s 435ms/step - loss: 1.5644 -
accuracy: 0.5669 - val_loss: 3.0322 - val_accuracy: 0.2143
Epoch 154/250
22/22 [==============================] - 10s 451ms/step - loss: 1.5335 -
accuracy: 0.6163 - val_loss: 3.0290 - val_accuracy: 0.2232
Epoch 155/250
22/22 [==============================] - 10s 453ms/step - loss: 1.5480 -
accuracy: 0.6047 - val_loss: 3.0154 - val_accuracy: 0.2232
Epoch 156/250
22/22 [==============================] - 10s 447ms/step - loss: 1.5152 -
accuracy: 0.6134 - val_loss: 3.0388 - val_accuracy: 0.2143
Epoch 157/250
22/22 [==============================] - 11s 502ms/step - loss: 1.5373 -
accuracy: 0.6076 - val_loss: 3.0236 - val_accuracy: 0.2232
Epoch 158/250
22/22 [==============================] - 11s 504ms/step - loss: 1.5004 -
accuracy: 0.6163 - val_loss: 3.0046 - val_accuracy: 0.2321
```

```
Epoch 159/250
22/22 [==============================] - 11s 500ms/step - loss: 1.5117 -
accuracy: 0.6192 - val_loss: 3.0153 - val_accuracy: 0.2054
Epoch 160/250
22/22 [==============================] - 9s 416ms/step - loss: 1.4834 -
accuracy: 0.6279 - val_loss: 3.0575 - val_accuracy: 0.2054
Epoch 161/250
22/22 [==============================] - 9s 390ms/step - loss: 1.5255 -
accuracy: 0.5988 - val_loss: 3.0919 - val_accuracy: 0.1696
Epoch 162/250
22/22 [==============================] - 10s 449ms/step - loss: 1.4959 -
accuracy: 0.6250 - val_loss: 3.0030 - val_accuracy: 0.1875
Epoch 163/250
22/22 [==============================] - 11s 503ms/step - loss: 1.4680 -
accuracy: 0.6279 - val_loss: 3.0897 - val_accuracy: 0.1875
Epoch 164/250
22/22 [==============================] - 9s 394ms/step - loss: 1.4874 -
accuracy: 0.6250 - val_loss: 2.9889 - val_accuracy: 0.2232
Epoch 165/250
22/22 [==============================] - 10s 443ms/step - loss: 1.4972 -
accuracy: 0.6483 - val_loss: 3.0249 - val_accuracy: 0.2143
Epoch 166/250
22/22 [==============================] - 10s 457ms/step - loss: 1.4579 -
accuracy: 0.6192 - val_loss: 3.0071 - val_accuracy: 0.2500
Epoch 167/250
22/22 [==============================] - 10s 457ms/step - loss: 1.4803 -
accuracy: 0.6424 - val_loss: 2.9320 - val_accuracy: 0.2589
Epoch 168/250
22/22 [==============================] - 10s 450ms/step - loss: 1.5327 -
accuracy: 0.5988 - val_loss: 3.0429 - val_accuracy: 0.2232
Epoch 169/250
22/22 [==============================] - 11s 478ms/step - loss: 1.4731 -
accuracy: 0.6453 - val_loss: 3.1124 - val_accuracy: 0.2143
Epoch 170/250
22/22 [==============================] - 11s 507ms/step - loss: 1.4558 -
accuracy: 0.6279 - val_loss: 3.0635 - val_accuracy: 0.2143
Epoch 171/250
22/22 [==============================] - 11s 503ms/step - loss: 1.4610 -
accuracy: 0.6337 - val_loss: 2.9628 - val_accuracy: 0.2321
Epoch 172/250
22/22 [==============================] - 9s 388ms/step - loss: 1.5010 -
accuracy: 0.5988 - val_loss: 3.0180 - val_accuracy: 0.2232
Epoch 173/250
22/22 [==============================] - 11s 509ms/step - loss: 1.4557 -
accuracy: 0.6105 - val_loss: 3.1083 - val_accuracy: 0.2232
Epoch 174/250
22/22 [==============================] - 10s 445ms/step - loss: 1.4232 -
accuracy: 0.6773 - val_loss: 3.0561 - val_accuracy: 0.2232
```

```
Epoch 175/250
22/22 [==============================] - 9s 405ms/step - loss: 1.4366 -
accuracy: 0.6424 - val_loss: 3.0175 - val_accuracy: 0.2321
Epoch 176/250
22/22 [==============================] - 9s 395ms/step - loss: 1.4080 -
accuracy: 0.6483 - val_loss: 3.0175 - val_accuracy: 0.2411
Epoch 177/250
22/22 [==============================] - 11s 497ms/step - loss: 1.4214 -
accuracy: 0.6279 - val_loss: 3.0224 - val_accuracy: 0.2500
Epoch 178/250
22/22 [==============================] - 11s 507ms/step - loss: 1.4430 -
accuracy: 0.6250 - val_loss: 3.1042 - val_accuracy: 0.2500
Epoch 179/250
22/22 [==============================] - 11s 498ms/step - loss: 1.4267 -
accuracy: 0.6512 - val_loss: 3.0606 - val_accuracy: 0.2500
Epoch 180/250
22/22 [==============================] - 9s 406ms/step - loss: 1.3608 -
accuracy: 0.6831 - val_loss: 3.0942 - val_accuracy: 0.2768
Epoch 181/250
22/22 [==============================] - 11s 471ms/step - loss: 1.4166 -
accuracy: 0.6424 - val_loss: 3.0232 - val_accuracy: 0.2500
Epoch 182/250
22/22 [==============================] - 11s 504ms/step - loss: 1.3914 -
accuracy: 0.6599 - val_loss: 3.0617 - val_accuracy: 0.2232
Epoch 183/250
22/22 [==============================] - 10s 471ms/step - loss: 1.3853 -
accuracy: 0.6541 - val_loss: 2.9177 - val_accuracy: 0.2411
Epoch 184/250
22/22 [==============================] - 10s 449ms/step - loss: 1.3761 -
accuracy: 0.6453 - val_loss: 2.9226 - val_accuracy: 0.2411
Epoch 185/250
22/22 [==============================] - 9s 396ms/step - loss: 1.3438 -
accuracy: 0.6773 - val_loss: 3.0275 - val_accuracy: 0.2589
Epoch 186/250
22/22 [==============================] - 11s 510ms/step - loss: 1.3576 -
accuracy: 0.6628 - val_loss: 2.9959 - val_accuracy: 0.2589
Epoch 187/250
22/22 [==============================] - 10s 442ms/step - loss: 1.4249 -
accuracy: 0.6337 - val_loss: 2.9558 - val_accuracy: 0.2589
Epoch 188/250
22/22 [==============================] - 9s 403ms/step - loss: 1.3754 -
accuracy: 0.6512 - val_loss: 3.0631 - val_accuracy: 0.2500
Epoch 189/250
22/22 [==============================] - 10s 445ms/step - loss: 1.3047 -
accuracy: 0.6744 - val_loss: 2.9412 - val_accuracy: 0.2946
Epoch 190/250
22/22 [==============================] - 10s 435ms/step - loss: 1.3464 -
accuracy: 0.6686 - val_loss: 3.0109 - val_accuracy: 0.2411
```

```
Epoch 191/250
22/22 [==============================] - 10s 448ms/step - loss: 1.3369 -
accuracy: 0.6831 - val_loss: 2.9519 - val_accuracy: 0.2589
Epoch 192/250
22/22 [==============================] - 10s 434ms/step - loss: 1.3555 -
accuracy: 0.6686 - val_loss: 3.0209 - val_accuracy: 0.2500
Epoch 193/250
22/22 [==============================] - 9s 396ms/step - loss: 1.3175 -
accuracy: 0.6860 - val_loss: 3.0225 - val_accuracy: 0.2589
Epoch 194/250
22/22 [==============================] - 11s 504ms/step - loss: 1.3274 -
accuracy: 0.6628 - val_loss: 2.9500 - val_accuracy: 0.2679
Epoch 195/250
22/22 [==============================] - 11s 512ms/step - loss: 1.3302 -
accuracy: 0.6860 - val_loss: 2.9612 - val_accuracy: 0.2589
Epoch 196/250
22/22 [==============================] - 10s 440ms/step - loss: 1.3240 -
accuracy: 0.6628 - val_loss: 2.9565 - val_accuracy: 0.2768
Epoch 197/250
22/22 [==============================] - 10s 458ms/step - loss: 1.2831 -
accuracy: 0.7122 - val_loss: 2.9646 - val_accuracy: 0.2768
Epoch 198/250
22/22 [==============================] - 10s 447ms/step - loss: 1.2975 -
accuracy: 0.6890 - val_loss: 3.0388 - val_accuracy: 0.2679
Epoch 199/250
22/22 [==============================] - 11s 504ms/step - loss: 1.3099 -
accuracy: 0.7006 - val_loss: 3.0254 - val_accuracy: 0.2679
Epoch 200/250
22/22 [==============================] - 11s 506ms/step - loss: 1.3382 -
accuracy: 0.6628 - val_loss: 2.9067 - val_accuracy: 0.2679
Epoch 201/250
22/22 [==============================] - 9s 410ms/step - loss: 1.3199 -
accuracy: 0.6773 - val_loss: 3.0187 - val_accuracy: 0.2411
Epoch 202/250
22/22 [==============================] - 11s 468ms/step - loss: 1.3114 -
accuracy: 0.6890 - val_loss: 2.9429 - val_accuracy: 0.2946
Epoch 203/250
22/22 [==============================] - 10s 453ms/step - loss: 1.2858 -
accuracy: 0.7035 - val_loss: 2.9890 - val_accuracy: 0.2768
Epoch 204/250
22/22 [==============================] - 10s 442ms/step - loss: 1.2744 -
accuracy: 0.7122 - val_loss: 3.0038 - val_accuracy: 0.2679
Epoch 205/250
22/22 [==============================] - 10s 441ms/step - loss: 1.2827 -
accuracy: 0.6890 - val_loss: 2.9606 - val_accuracy: 0.2857
Epoch 206/250
22/22 [==============================] - 10s 444ms/step - loss: 1.2973 -
accuracy: 0.6860 - val_loss: 3.0803 - val_accuracy: 0.2679
```

```
Epoch 207/250
22/22 [==============================] - 11s 503ms/step - loss: 1.2532 -
accuracy: 0.7093 - val_loss: 3.0766 - val_accuracy: 0.2857
Epoch 208/250
22/22 [==============================] - 10s 450ms/step - loss: 1.2452 -
accuracy: 0.7035 - val_loss: 2.9661 - val_accuracy: 0.2857
Epoch 209/250
22/22 [==============================] - 10s 467ms/step - loss: 1.2420 -
accuracy: 0.7093 - val_loss: 2.9406 - val_accuracy: 0.3304
Epoch 210/250
22/22 [==============================] - 10s 462ms/step - loss: 1.2593 -
accuracy: 0.7035 - val_loss: 3.0327 - val_accuracy: 0.2857
Epoch 211/250
22/22 [==============================] - 11s 499ms/step - loss: 1.2551 -
accuracy: 0.7064 - val_loss: 2.8546 - val_accuracy: 0.3036
Epoch 212/250
22/22 [==============================] - 10s 441ms/step - loss: 1.2626 -
accuracy: 0.7093 - val_loss: 3.0098 - val_accuracy: 0.2679
Epoch 213/250
22/22 [==============================] - 9s 393ms/step - loss: 1.2843 -
accuracy: 0.7006 - val_loss: 2.9970 - val_accuracy: 0.2589
Epoch 214/250
22/22 [==============================] - 10s 450ms/step - loss: 1.2423 -
accuracy: 0.7064 - val_loss: 3.1198 - val_accuracy: 0.2143
Epoch 215/250
22/22 [==============================] - 10s 445ms/step - loss: 1.2215 -
accuracy: 0.7006 - val_loss: 3.0243 - val_accuracy: 0.2589
Epoch 216/250
22/22 [==============================] - 9s 408ms/step - loss: 1.2167 -
accuracy: 0.7244 - val_loss: 3.0162 - val_accuracy: 0.2589
Epoch 217/250
22/22 [==============================] - 10s 446ms/step - loss: 1.2286 -
accuracy: 0.7188 - val_loss: 3.0475 - val_accuracy: 0.2679
Epoch 218/250
22/22 [==============================] - 11s 487ms/step - loss: 1.2294 -
accuracy: 0.7180 - val_loss: 3.0355 - val_accuracy: 0.2857
Epoch 219/250
22/22 [==============================] - 11s 506ms/step - loss: 1.2061 -
accuracy: 0.6977 - val_loss: 2.9468 - val_accuracy: 0.2857
Epoch 220/250
22/22 [==============================] - 9s 435ms/step - loss: 1.2513 -
accuracy: 0.6948 - val_loss: 3.0215 - val_accuracy: 0.2946
Epoch 221/250
22/22 [==============================] - 9s 397ms/step - loss: 1.2389 -
accuracy: 0.7006 - val_loss: 3.0412 - val_accuracy: 0.2679
Epoch 222/250
22/22 [==============================] - 11s 501ms/step - loss: 1.2430 -
accuracy: 0.6948 - val_loss: 2.9764 - val_accuracy: 0.2679
```

```
Epoch 223/250
22/22 [==============================] - 10s 446ms/step - loss: 1.2390 -
accuracy: 0.6948 - val_loss: 3.0118 - val_accuracy: 0.2679
Epoch 224/250
22/22 [==============================] - 10s 466ms/step - loss: 1.2147 -
accuracy: 0.7093 - val_loss: 3.0551 - val_accuracy: 0.2143
Epoch 225/250
22/22 [==============================] - 10s 446ms/step - loss: 1.2403 -
accuracy: 0.7122 - val_loss: 3.0965 - val_accuracy: 0.2321
Epoch 226/250
22/22 [==============================] - 9s 414ms/step - loss: 1.2186 -
accuracy: 0.7151 - val_loss: 3.1221 - val_accuracy: 0.2321
Epoch 227/250
22/22 [==============================] - 10s 443ms/step - loss: 1.2304 -
accuracy: 0.7035 - val_loss: 3.0494 - val_accuracy: 0.2679
Epoch 228/250
22/22 [==============================] - 10s 444ms/step - loss: 1.1584 -
accuracy: 0.7297 - val_loss: 3.0801 - val_accuracy: 0.2589
Epoch 229/250
22/22 [==============================] - 10s 457ms/step - loss: 1.1934 -
accuracy: 0.7355 - val_loss: 3.0798 - val_accuracy: 0.2589
Epoch 230/250
22/22 [==============================] - 9s 387ms/step - loss: 1.2125 -
accuracy: 0.7064 - val_loss: 3.0176 - val_accuracy: 0.2768
Epoch 231/250
22/22 [==============================] - 11s 503ms/step - loss: 1.1782 -
accuracy: 0.6977 - val_loss: 3.0727 - val_accuracy: 0.2679
Epoch 232/250
22/22 [==============================] - 10s 445ms/step - loss: 1.1577 -
accuracy: 0.7442 - val_loss: 3.0207 - val_accuracy: 0.2589
Epoch 233/250
22/22 [==============================] - 9s 403ms/step - loss: 1.2093 -
accuracy: 0.7180 - val_loss: 2.9969 - val_accuracy: 0.2589
Epoch 234/250
22/22 [==============================] - 9s 382ms/step - loss: 1.1662 -
accuracy: 0.7267 - val_loss: 3.0008 - val_accuracy: 0.2857
Epoch 235/250
22/22 [==============================] - 9s 419ms/step - loss: 1.1918 -
accuracy: 0.6977 - val_loss: 3.0319 - val_accuracy: 0.2768
Epoch 236/250
22/22 [==============================] - 10s 446ms/step - loss: 1.1859 -
accuracy: 0.7006 - val_loss: 3.0010 - val_accuracy: 0.2679
Epoch 237/250
22/22 [==============================] - 10s 478ms/step - loss: 1.1403 -
accuracy: 0.7587 - val_loss: 2.9705 - val_accuracy: 0.2946
Epoch 238/250
22/22 [==============================] - 9s 394ms/step - loss: 1.1526 -
accuracy: 0.7386 - val_loss: 3.0950 - val_accuracy: 0.2500
```

```
Epoch 239/250
22/22 [==============================] - 11s 498ms/step - loss: 1.1703 -
accuracy: 0.7209 - val_loss: 3.0436 - val_accuracy: 0.2321
Epoch 240/250
22/22 [==============================] - 10s 448ms/step - loss: 1.1694 -
accuracy: 0.7209 - val_loss: 3.0996 - val_accuracy: 0.2500
Epoch 241/250
22/22 [==============================] - 10s 460ms/step - loss: 1.1534 -
accuracy: 0.7355 - val_loss: 2.9982 - val_accuracy: 0.2679
Epoch 242/250
22/22 [==============================] - 8s 377ms/step - loss: 1.2006 -
accuracy: 0.7151 - val_loss: 3.0675 - val_accuracy: 0.2232
Epoch 243/250
22/22 [==============================] - 11s 503ms/step - loss: 1.1224 -
accuracy: 0.7384 - val_loss: 3.0261 - val_accuracy: 0.2411
Epoch 244/250
22/22 [==============================] - 10s 448ms/step - loss: 1.1124 -
accuracy: 0.7413 - val_loss: 3.0897 - val_accuracy: 0.2500
Epoch 245/250
22/22 [==============================] - 9s 415ms/step - loss: 1.1277 -
accuracy: 0.7442 - val_loss: 3.0283 - val_accuracy: 0.2500
Epoch 246/250
22/22 [==============================] - 9s 388ms/step - loss: 1.1469 -
accuracy: 0.7442 - val_loss: 2.9735 - val_accuracy: 0.2768
Epoch 247/250
22/22 [==============================] - 11s 471ms/step - loss: 1.1462 -
accuracy: 0.7297 - val_loss: 2.9964 - val_accuracy: 0.2946
Epoch 248/250
22/22 [==============================] - 10s 438ms/step - loss: 1.1376 -
accuracy: 0.7209 - val_loss: 3.1330 - val_accuracy: 0.2589
Epoch 249/250
22/22 [==============================] - 11s 494ms/step - loss: 1.0669 -
accuracy: 0.7558 - val_loss: 3.0366 - val_accuracy: 0.2768
Epoch 250/250
22/22 [==============================] - 9s 397ms/step - loss: 1.1195 -
accuracy: 0.7587 - val_loss: 3.0153 - val_accuracy: 0.2946
```

[ ]:

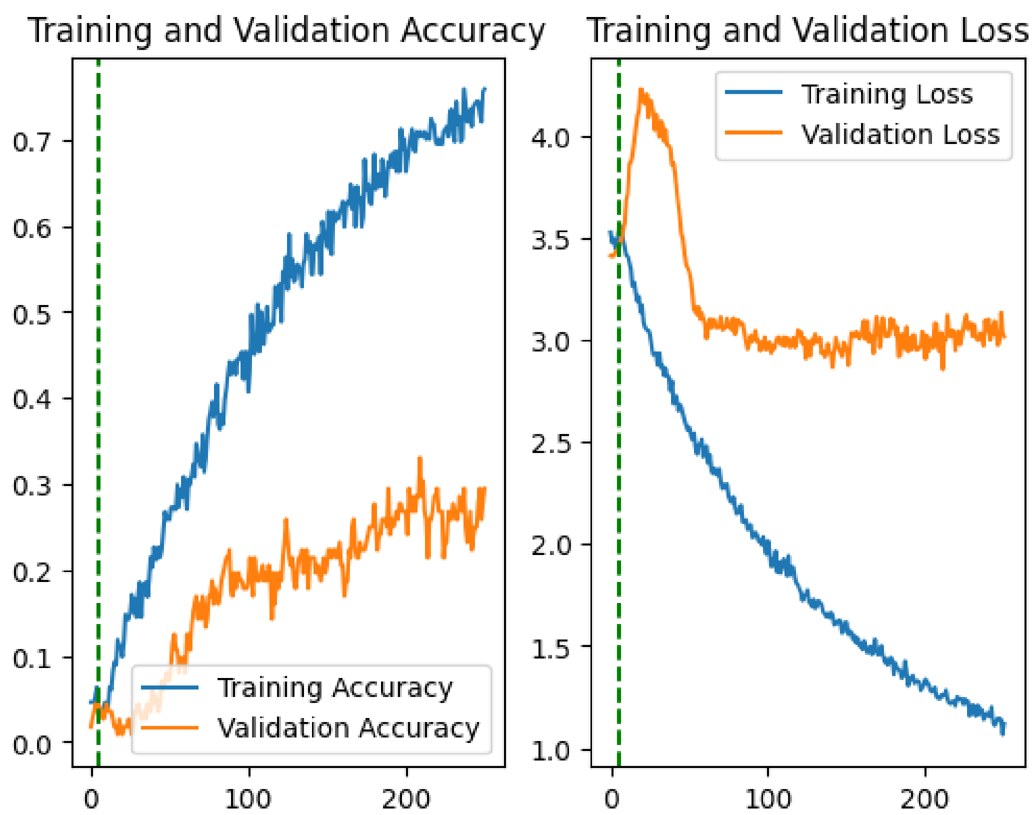
```
# Plot training history
import matplotlib.pyplot as plt

acc = history.history['accuracy'] + history_fine.history['accuracy']
val_acc = history.history['val_accuracy'] + history_fine.history['val_accuracy']
loss = history.history['loss'] + history_fine.history['loss']
val_loss = history.history['val_loss'] + history_fine.history['val_loss']
epochs_range = range(total_epochs+1)

plt.subplot(1, 2, 1)
```

```
plt.plot(epochs_range, acc, label='Training Accuracy')
plt.plot(epochs_range, val_acc, label='Validation Accuracy')
plt.axvline(x=len(history.history['accuracy']), color='green', linestyle='--')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

plt.subplot(1, 2, 2)
plt.plot(epochs_range, loss, label='Training Loss')
plt.plot(epochs_range, val_loss, label='Validation Loss')
plt.axvline(x=len(history.history['accuracy']), color='green', linestyle='--')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()
```



Based on the above training and graphs, which depicts the training and validation accuracy and its losses, over epochs, here is the discussion:

1. Performance of the network: The training accuracy shows a steady upward trend, signaling consistent improvement of the network's performance on the training dataset. Conversely, the growth of the validation accuracy is more gradual and plateaus toward the later epochs. This suggests that the model may be overfitting to the training data and, consequently, is not performing as well on data it has not seen before.

2. Convergence: The training loss experiences a rapid decline before stabilizing, which indicates that the model is effectively learning and stabilizing with the training data. The validation loss initially drops but then begins to show variability, pointing to a more gradual and less stable convergence concerning the unseen data.

3. Hyperparameters tweaks tried:

- Initially, attempts to train a VGG16 model did not yield an increase in accuracy for either the training or validation datasets, suggesting that the model wasn't effectively learning from the data.
- Subsequently, the switch to a ResNet50 model resulted in an uptick in training accuracy, which was a positive development, though the validation accuracy remained stagnant.
- Various batch sizes, including 12, 16 and 32, were experimented with; it was observed that a batch size of 16 led to a steady improvement in the model's training accuracy.
- The implementation of a learning rate scheduler and ReduceLROnPlateau callbacks slowed down the training, and the accuracy gains relative to the number of epochs were marginal.
- The model's architecture was then adjusted to include dropout and batch normalization layers, and the dropout rate was fine-tuned. These adjustments significantly enhanced the network's training capacity, culminating in improved accuracy and reduced loss on the training dataset, alongside better accuracy and loss metrics on the validation set.
- To further refine the network's performance, there's the potential to add more Dense layers, incorporate additional dropout layers, adjust dropout rates, and prolong the training duration across more epochs. Such enhancements aim to harness the intricate architecture of ResNet50 and elevate the model's capabilities. This process will likely entail extensive training and meticulous hyperparameter optimization.

4. Conclusion on training: The presence of a bottleneck layer with only 8 neurons preceding the final output layer of 30 neurons could be restricting the model's learning capacity. This narrow layer may be limiting the flow of information, rendering the knowledge acquired by these 8 neurons insufficient for the demands of the subsequent 30-neuron layer, thereby resulting in sluggish learning and initial underperformance of the model. To enhance the model's learning, increasing the number of neurons in the bottleneck layer, extending the number of training epochs, and adding complexity to the model may be beneficial strategies to pursue.

4. Compute the correlation coefficients between the network data and human data for each of the 8 neurons in your next-to-last layer (similar to HW3, using procrustes analysis) for the images from the train set (360 images) and (separately) for the images from the validation set (120 images). Report each of the 8 correlation coefficients and your average correlation coefficients (please mark in the bold with large font so we can easily find it).

[ ]:
```
import os
import numpy as np
from tensorflow.keras.preprocessing import image

def createNumpyArrayOfImages(dir):
  images = []
  for category in sorted(os.listdir(dir)):
```

```
        if not os.path.isdir(os.path.join(dir, category)):
            continue

        for img_file in sorted(os.listdir(os.path.join(dir, category))):
            img_path = os.path.join(dir, category, img_file)
            img = image.load_img(img_path, target_size=(224, 224))
            img_array = image.img_to_array(img)
            img_array = np.expand_dims(img_array, axis=0)
            img_array = img_array / 255.
            images.append(img_array)
    images = np.vstack(images)

    return images

train_dir = '/content/drive/My Drive/training_dataset'
val_dir = '/content/drive/My Drive/validation_dataset'


train_images = createNumpyArrayOfImages(train_dir)
val_images = createNumpyArrayOfImages(val_dir)
```

[ ]:
```
model_extract = Model(inputs=model.input, outputs=model.layers[-2].output)
train_activations = model_extract.predict(train_images)
val_activations = model_extract.predict(val_images)
```

```
12/12 [==============================] - 5s 81ms/step
4/4 [==============================] - 2s 672ms/step
```

[ ]:
```
with open('/content/drive/My Drive/mds_360.txt', 'r') as file:
    human_data_train = np.array([[float(num) for num in line.split()] for line␣
 ↪in file])

with open('/content/drive/My Drive/mds_120.txt', 'r') as file:
    human_data_val = np.array([[float(num) for num in line.split()] for line in␣
 ↪file])
```

[ ]:
```
train_activations.shape
```

[ ]: (360, 8)

[ ]:
```
human_data_train.shape
```

[ ]: (360, 8)

The train_activations and human_data_train are in the shape of (360, 8) and for the val_activations and human_data_val are in the shape of (120, 8)

```
[ ]: from sklearn.preprocessing import StandardScaler

     scaler = StandardScaler()

     train_activations_norm = scaler.fit_transform(train_activations)
     human_data_train_norm = scaler.fit_transform(human_data_train)

     val_activations_norm = scaler.transform(val_activations)
     human_data_val_norm = scaler.transform(human_data_val)
```

```
[ ]: from scipy.spatial import procrustes

     train_mtx1, train_mtx2, disparity_train = procrustes(human_data_train_norm,␣
      ↪train_activations_norm)
     val_mtx1, val_mtx2, disparity_val = procrustes(human_data_val_norm,␣
      ↪val_activations_norm)
```

```
[ ]: from scipy.stats import pearsonr

     train_correlations = [pearsonr(train_mtx1[:,i], train_mtx2[:,i])[0] for i in␣
      ↪range(train_mtx2.shape[1])]
     val_correlations = [pearsonr(val_mtx1[:,i], val_mtx2[:,i])[0] for i in␣
      ↪range(val_mtx2.shape[1])]
```

```
[ ]: train_correlations[:5]
```

```
[ ]: [0.3164635221648821,
      0.29386582663786975,
      0.1688508555213972,
      0.45899654223289965,
      0.3502391980235649]
```

```
[ ]: val_correlations[:5]
```

```
[ ]: [0.4751742757880976,
      0.2805675036809753,
      0.3539393679740733,
      0.5550301707266293,
      0.43851298572007524]
```

```
[ ]: average_correlation_train = np.mean(train_correlations)
     average_correlation_val = np.mean(val_correlations)

     print(f"Average Correlation Coefficients for Training Set:␣
      ↪{average_correlation_train:.3f}")
     print(f"Average Correlation Coefficients for Validation Set:␣
      ↪{average_correlation_val:.3f}")
```

```
Average Correlation Coefficients for Training Set: 0.312
Average Correlation Coefficients for Validation Set: 0.442
```

[ ]:

```

```