COVID-19-Image-Classification-phase1

November 24, 2021

1 DS6050 Deep Learning Project

Project: Covid19 Image Classification - Phase 1 Team: Paul Hicks(pdh2d), Sudharshan Luthra(sl3zs), Jay Hombal(mh4ey), Matt Dakolios(mrd7f)

Abstract: Our Aim is to detect Covid19 from chest X-rays. The covid19 image dataset we are using is small with about 3000 classes belonging to three classes 'Normal', 'Covid19' and 'Pneumonia' respectively. This dataset small and is insufficient to generalize. So for the purpoe of our project, in Phase-I we will first use NIH X-ray image data to retrain and finetune pretrained model architecture such as ResNet50V2, MobileNetV2 and VGG16.

In Phase2, we intend to reload the best saved model from Phase 1 to train, validate, finetune the model and finally evaluate the classifications on the target covid19 dataset.

Reference: 1. https://www.kaggle.com/nih-chest-xrays/data 2. https://www.kaggle.com/mushaxyz/covid19-customized-xray-dataset

1.1 Project Code Orginzation

Cookiecutter is a command-line utility that creates projects from cookiecutters (project templates), e.g. Python package projects, LaTeX documents, etc.

Installed and created the project template using Cookiecutter:
Follow instructions from https://ericbassett.tech/cookiecutter-data-science-crash-course/

1.2 Validating and pre-processing NIH X-ray metadata dataset

Following instructions use make tool, run commands from from your terminal from your project folder

Setup
Setup python environment

```
1. Validate Python is installed and create required directories
    Run: make test_environment
```

Data Extraction:(execute only once)

```
2. Download and unzip the NIH X-ray images in data/raw
    Run: make get_nih_images
```

Data Validation:(execute only once)

```
3. Validate Dataset (rename columns and delete patient record with age greater than 100)
    Run: make validate_nih_images
```

Data Prepartion:(execute only once)

```
4. Prepare Dataset (add path attribute, split dataset into train and validation dataset)
    Run: make prepare_nih_images
```

This proudces the three output files in processed folder: 1. prepared_data_entry_2017.csv (full dataset) 2. prepared_train_data_entry_2017.csv (train_dataset) 3. prepared_valid_data_entry_2017.csv (validation_dataset)

Next, we use prepared_train_data_entry_2017.csv and prepared_valid_data_entry_2017.csv files to retrain CNN model architectures pre-trained using IMAGENET database

1.3 Imports and Setup

[1]:

```python
# common imports
import os
import numpy as np
import matplotlib.pylab as plt
import pandas as pd
import random
from glob import glob
from pathlib import Path
from functools import partial
from sklearn.model_selection import GroupShuffleSplit
from sklearn.model_selection import train_test_split

os.environ["TF_CPP_MIN_LOG_LEVEL"] = "2"
# prevent VRAM occupied
os.environ['TF_FORCE_GPU_ALLOW_GROWTH'] = 'true'

# TensorFlow  2.0 is required
import tensorflow as tf
from tensorflow import keras
assert tf.__version__ >= "2.0"

import warnings
warnings.filterwarnings('ignore')
```

#Change working directory - as the images are located in data/raw in the project folder **(Execute this cell Once)**

[2]:

```python
if '/notebooks' in os.getcwd():
    os.chdir("../")
    print("set the project directory as working directory")
else:
    print(os.getcwd())
```

set the project directory as working directory

[3]:

```
# Import functions for trianing the model
%load_ext autoreload
%autoreload 2
import src.models.train_model as train_model

# load tensorboard extension
%reload_ext tensorboard
```

1.4 Data Ingestion

[4]:

```
# Constants
SEED =42
IMAGE_SIZE = (224,224)
IMAGE_SHAPE = (224,224,3)
BATCH_SIZE = 32
SHUFFLE = True
NUM_CLASSES = 15 # number of ClassesNUM
NUM_EPOCHS = 10
PRETRAINED_MODELS = ['ResNet50V2', 'MobileNetV2', 'VGG16']

# Train and validate function
def train_and_validate_model(model_name,
                             train_generator,
                             valid_generator,
                             save_model_filepath: str,
                             logs_dir: str,
                             freeze_layers:bool = True,
                             activation: str = 'softmax',
                             learning_rate: float =0.01,
                             fine_tune_learning_rate: float = 0.0001,
                             fine_tune_at_layer:int = 186,
                             num_epochs:int = NUM_EPOCHS,
                             num_classes: int = NUM_CLASSES,
                             batch_size: int = BATCH_SIZE,
                             input_shape: int = IMAGE_SHAPE):

    print(model_name)

    my_model = train_model.
 ↪get_base_model_with_new_toplayer(base_model=model_name,
                                                         freeze_layers =␣
 ↪freeze_layers,
                                                         num_classes =␣
 ↪num_classes,
                                                         ␣
 ↪activation_func=activation,
                                                          learning_rate =␣
 ↪learning_rate,
                                                          input_shape =␣
 ↪input_shape)

    my_model_history = train_model.fit_model(my_model,
                                             train_generator,
                                             valid_generator,
                                             num_epochs=num_epochs,
                                             batch_size=batch_size,
                                            ␣
 ↪checkpoint_filepath=save_model_filepath,
                                             logs_dir = logs_dir)

    print(f'{model_name} Accuracy and Loss plots')
    train_model.plot_accuracy_and_loss(my_model_history)


    print("\n")
    #fine_tune model_name
    model_ft = train_model.fine_tune_model(my_model,
                                           fine_tune_learning_rate,
                                           optimizer='Adam',
                                          ␣
 ↪fine_tune_at_layer=fine_tune_at_layer,
                                           activation_func=activation)


    print("\n")
    print(f'Fine-Tuned {model_name} Training and Validation: ')
    model_ft_history = train_model.fit_model(model_ft, train_generator,
            valid_generator, num_epochs=num_epochs,batch_size=batch_size)
    print(f'Fine-Tuned {model_name} Accuracy and Loss plots')
    train_model.plot_accuracy_and_loss(model_ft_history)
    return model_ft
```

[5]:

```
def load_data():
    nih_xrays_train_df = pd.read_csv('data/processed/
 ↪prepared_train_data_entry_2017.csv')
    nih_xrays_valid_df = pd.read_csv('data/processed/
 ↪prepared_valid_data_entry_2017.csv')
    return nih_xrays_train_df,nih_xrays_valid_df
nih_xrays_train_df, nih_xrays_valid_df = load_data()
```

[6]:

```
# Get fourteen unique diagnosis
# It is a function that takes a series of iterables and returns one iterable
# The asterisk "*" is used in Python to define a variable number of arguments.
# The asterisk character has to precede a variable identifier in the parameter␣
 ↪list
from itertools import chain
all_labels = np.unique(list(chain(*nih_xrays_train_df['finding_label'].
 ↪map(lambda x: x.split('|')).tolist())))
# remove the empty label
all_labels = [x for x in all_labels if len(x)>0]
print('All Labels ({}): {}'.format(len(all_labels), all_labels))
```

```
All Labels (15): ['Atelectasis', 'Cardiomegaly', 'Consolidation', 'Edema',
'Effusion', 'Emphysema', 'Fibrosis', 'Hernia', 'Infiltration', 'Mass',
'NoFinding', 'Nodule', 'Pleural_Thickening', 'Pneumonia', 'Pneumothorax']
```

1.5 Preprocess Images

[7]:

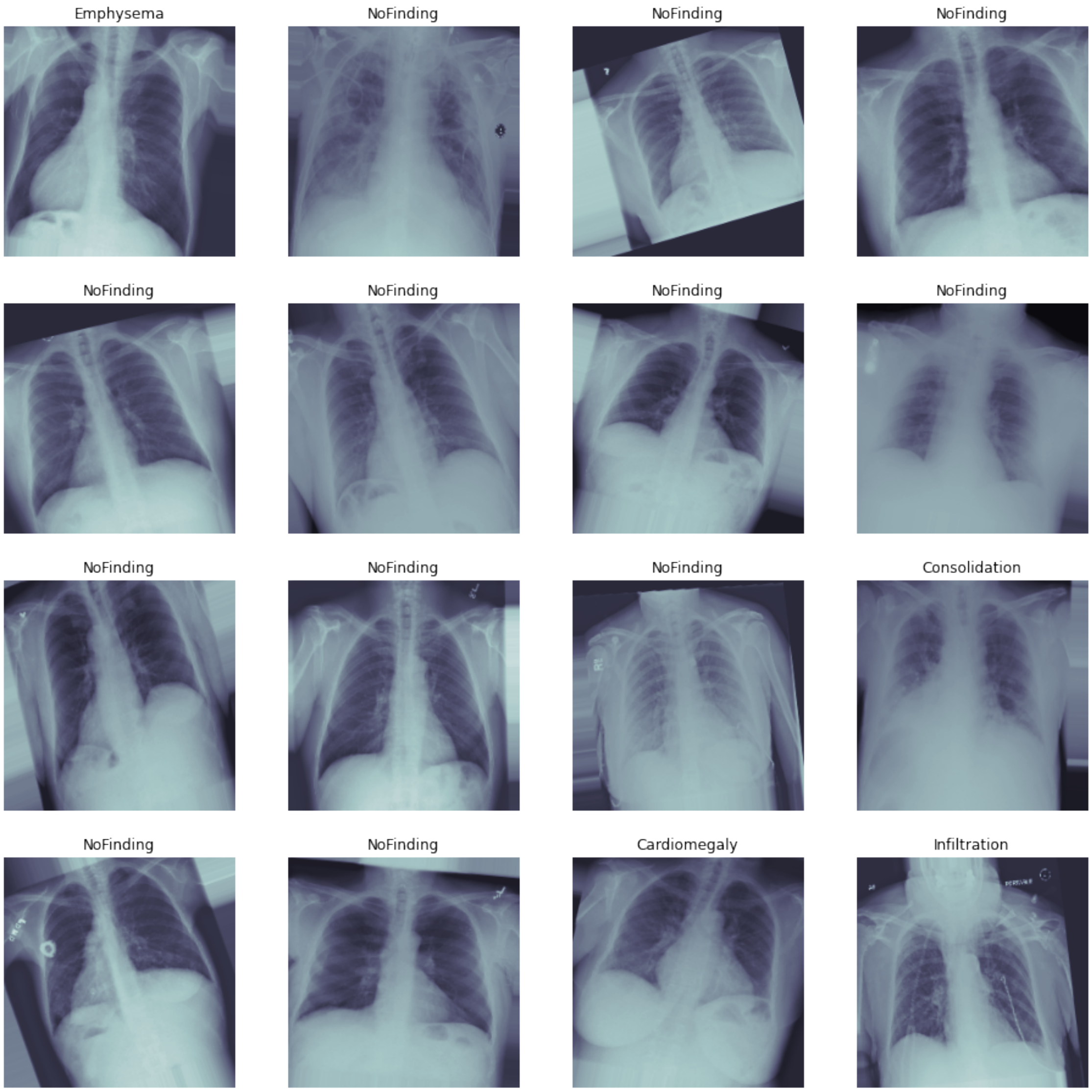
```
from keras.applications.resnet_v2 import preprocess_input
from keras.preprocessing.image import ImageDataGenerator

train_generator = train_model.
 ↪get_image_data_generator(nih_xrays_train_df,batch_size=BATCH_SIZE,image_size=IMAGE_SIZE,lab
valid_generator = train_model.
 ↪get_image_data_generator(nih_xrays_train_df,batch_size=BATCH_SIZE,image_size=IMAGE_SIZE,lab
```

```
Found 73141 validated image filenames belonging to 15 classes.
Found 73141 validated image filenames belonging to 15 classes.
```

1.5.1 Visualize Images

[8]:

```
t_x, t_y = next(train_generator)
fig, m_axs = plt.subplots(4, 4, figsize = (16, 16))
for (c_x, c_y, c_ax) in zip(t_x, t_y, m_axs.flatten()):
    c_ax.imshow(c_x[:,:,0], cmap = 'bone', vmin = -1.5, vmax = 1.5)
    c_ax.set_title(', '.join([n_class for n_class, n_score in zip(all_labels,␣
 ↪c_y)
                             if n_score>0.5]))
    c_ax.axis('off')
```



1.6 Experiment 1: Classification using all NIH data

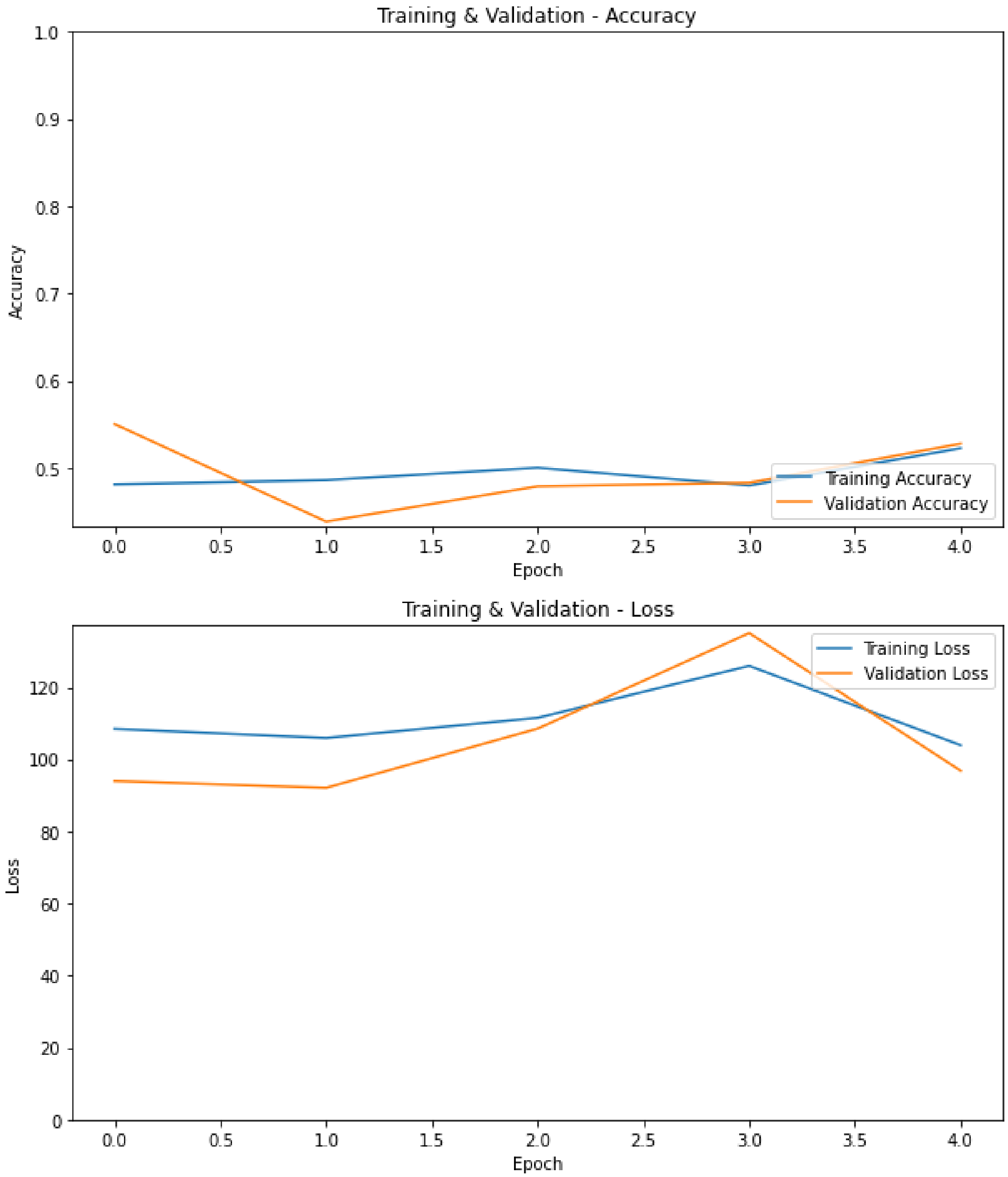
train_model.py model includes all functions for training the model (src/models/train_model.py)

1.6.1 ResNetV250

[9]:

```
model_name = PRETRAINED_MODELS[0]
save_model_filepath = 'models/'+ model_name + 'exp1.h5'
logs_dir = 'logs/fit/ResNet50V2exp1'
model = train_and_validate_model(model_name = model_name,
                                 train_generator=train_generator,
                                 valid_generator=valid_generator,
                                 save_model_filepath=save_model_filepath,
                                 logs_dir=logs_dir)
```

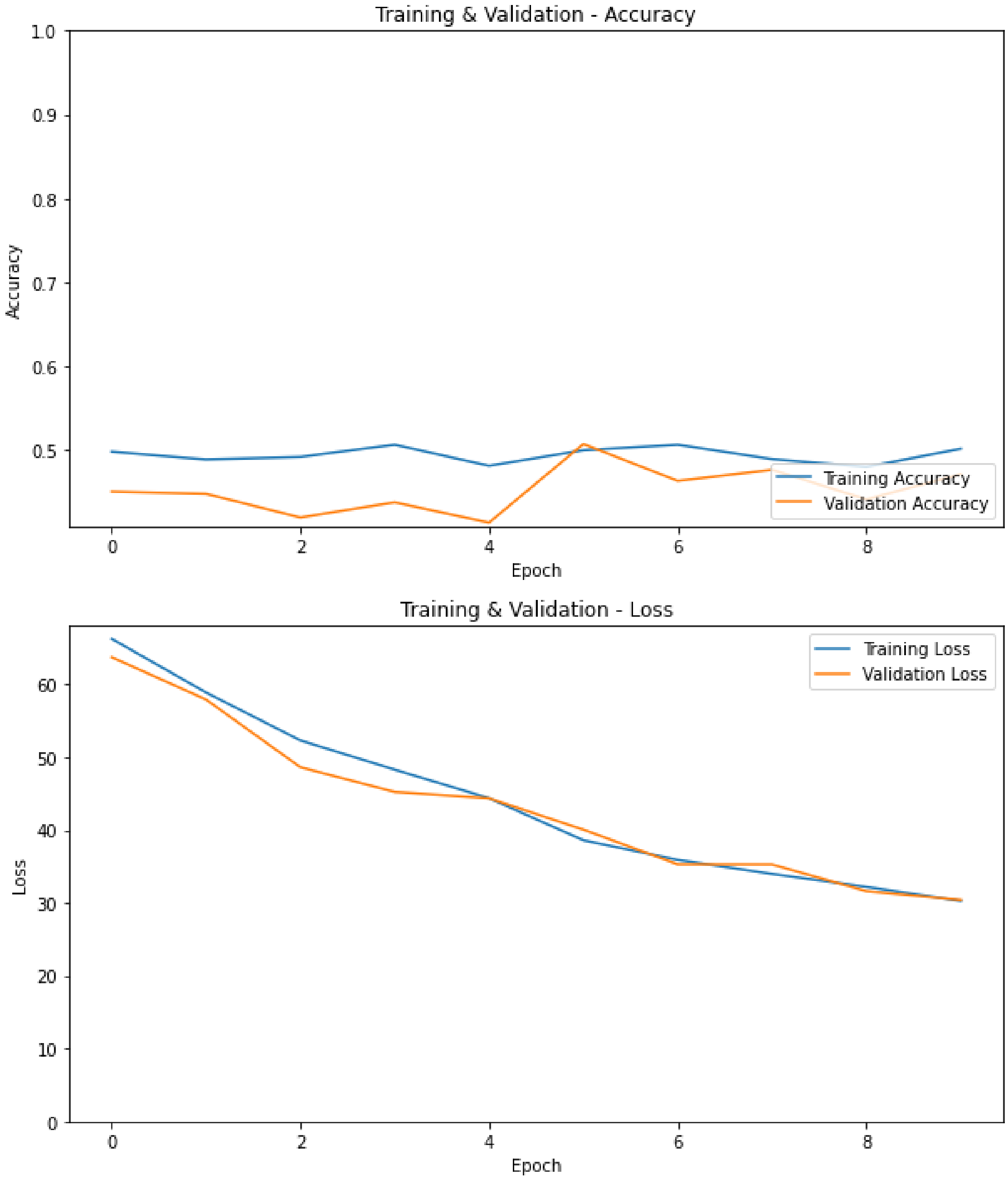
```
ResNet50V2
learning rate 0.01
Downloading ResNet50V2
Epoch 1/10
71/71 [==============================] - 71s 980ms/step - loss: 108.4804 -
accuracy: 0.4811 - val_loss: 93.9877 - val_accuracy: 0.5502

Epoch 00001: val_loss improved from inf to 93.98772, saving model to
models/ResNet50V2exp1.h5
Epoch 2/10
71/71 [==============================] - 68s 967ms/step - loss: 105.9499 -
accuracy: 0.4859 - val_loss: 92.1020 - val_accuracy: 0.4384

Epoch 00002: val_loss improved from 93.98772 to 92.10203, saving model to
models/ResNet50V2exp1.h5
Epoch 3/10
71/71 [==============================] - 68s 970ms/step - loss: 111.5486 -
accuracy: 0.5000 - val_loss: 108.5482 - val_accuracy: 0.4789

Epoch 00003: val_loss did not improve from 92.10203
Epoch 4/10
71/71 [==============================] - 69s 977ms/step - loss: 125.9536 -
accuracy: 0.4798 - val_loss: 135.0746 - val_accuracy: 0.4828

Epoch 00004: val_loss did not improve from 92.10203
Epoch 5/10
71/71 [==============================] - 68s 969ms/step - loss: 103.9409 -
accuracy: 0.5224 - val_loss: 96.8459 - val_accuracy: 0.5277

Epoch 00005: val_loss did not improve from 92.10203
Restoring model weights from the end of the best epoch.
Epoch 00005: early stopping
ResNet50V2 Accuracy and Loss plots
```



```
Fine-Tuned ResNet50V2 Training and Validation:
Epoch 1/10
71/71 [==============================] - 70s 975ms/step - loss: 66.2585 -
accuracy: 0.4982 - val_loss: 63.7342 - val_accuracy: 0.4507
```

```
Epoch 00001: val_loss improved from inf to 63.73417, saving model to
models/my_model.h5
Epoch 2/10
71/71 [==============================] - 68s 958ms/step - loss: 58.8906 -
accuracy: 0.4890 - val_loss: 57.9251 - val_accuracy: 0.4481

Epoch 00002: val_loss improved from 63.73417 to 57.92511, saving model to
models/my_model.h5
Epoch 3/10
71/71 [==============================] - 68s 965ms/step - loss: 52.3125 -
accuracy: 0.4921 - val_loss: 48.6318 - val_accuracy: 0.4199

Epoch 00003: val_loss improved from 57.92511 to 48.63180, saving model to
models/my_model.h5
Epoch 4/10
71/71 [==============================] - 68s 961ms/step - loss: 48.2973 -
accuracy: 0.5066 - val_loss: 45.2377 - val_accuracy: 0.4379

Epoch 00004: val_loss improved from 48.63180 to 45.23765, saving model to
models/my_model.h5
Epoch 5/10
71/71 [==============================] - 68s 967ms/step - loss: 44.3842 -
accuracy: 0.4815 - val_loss: 44.3694 - val_accuracy: 0.4137

Epoch 00005: val_loss improved from 45.23765 to 44.36945, saving model to
models/my_model.h5
Epoch 6/10
71/71 [==============================] - 68s 958ms/step - loss: 38.6037 -
accuracy: 0.5000 - val_loss: 40.0627 - val_accuracy: 0.5075

Epoch 00006: val_loss improved from 44.36945 to 40.06271, saving model to
models/my_model.h5
Epoch 7/10
71/71 [==============================] - 67s 955ms/step - loss: 35.9082 -
accuracy: 0.5066 - val_loss: 35.2999 - val_accuracy: 0.4635

Epoch 00007: val_loss improved from 40.06271 to 35.29994, saving model to
models/my_model.h5
Epoch 8/10
71/71 [==============================] - 68s 960ms/step - loss: 33.9838 -
accuracy: 0.4894 - val_loss: 35.2880 - val_accuracy: 0.4767

Epoch 00008: val_loss improved from 35.29994 to 35.28795, saving model to
models/my_model.h5
Epoch 9/10
71/71 [==============================] - 68s 958ms/step - loss: 32.1974 -
accuracy: 0.4802 - val_loss: 31.6149 - val_accuracy: 0.4415
```

```
Epoch 00009: val_loss improved from 35.28795 to 31.61493, saving model to
models/my_model.h5
Epoch 10/10
71/71 [==============================] - 67s 954ms/step - loss: 30.2840 -
accuracy: 0.5018 - val_loss: 30.4199 - val_accuracy: 0.4714

Epoch 00010: val_loss improved from 31.61493 to 30.41992, saving model to
models/my_model.h5
Fine-Tuned ResNet50V2 Accuracy and Loss plots
```



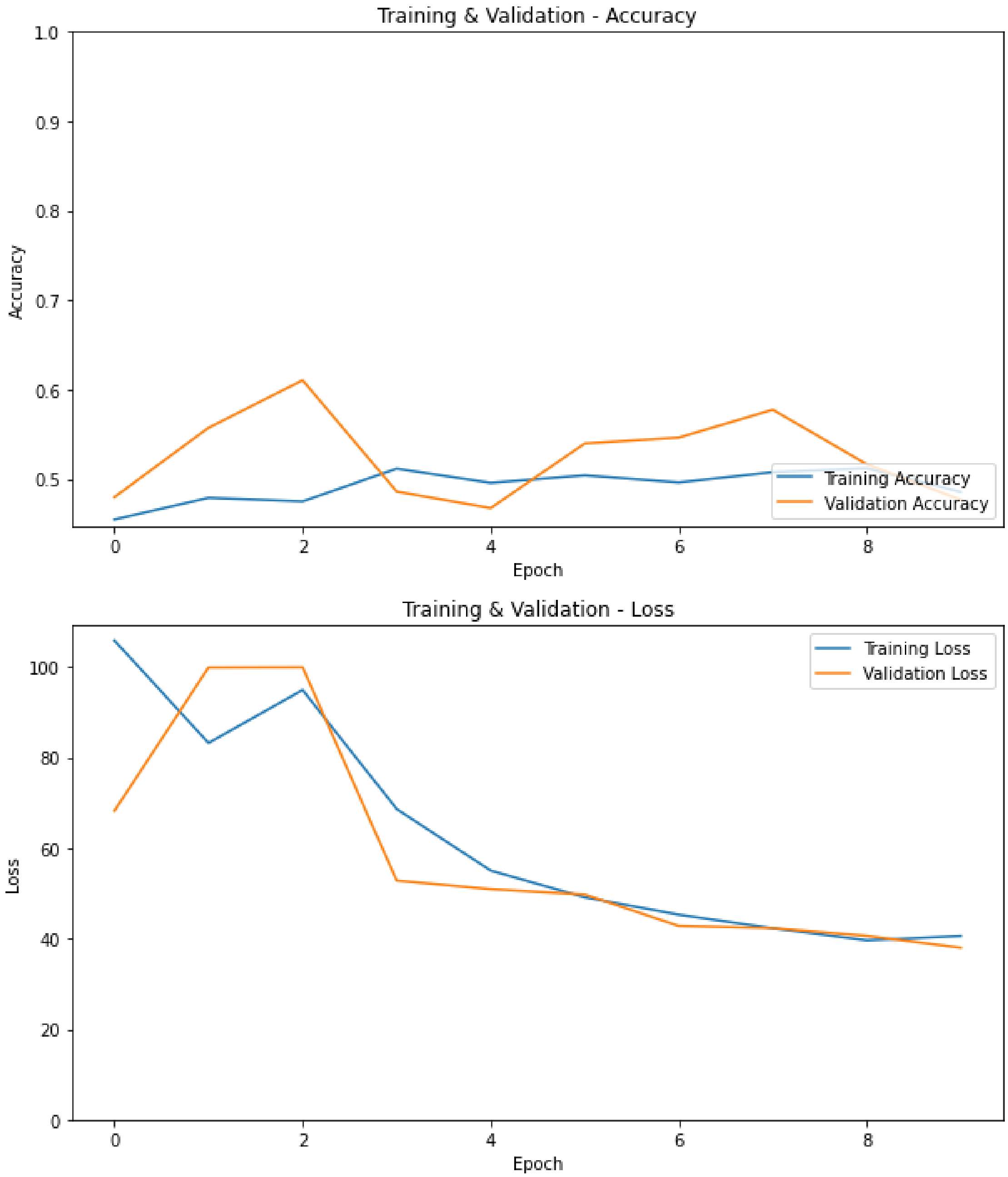
1.6.2 MobileNetV2

[10]:

```
model_name = PRETRAINED_MODELS[1]
save_model_filepath = 'models/'+ model_name + 'exp1.h5'
logs_dir = 'logs/fit/MobileNetV2exp1'
model = train_and_validate_model(model_name = model_name,
                                 train_generator=train_generator,
                                 valid_generator=valid_generator,
                                 save_model_filepath=save_model_filepath,
                                 logs_dir=logs_dir)
```

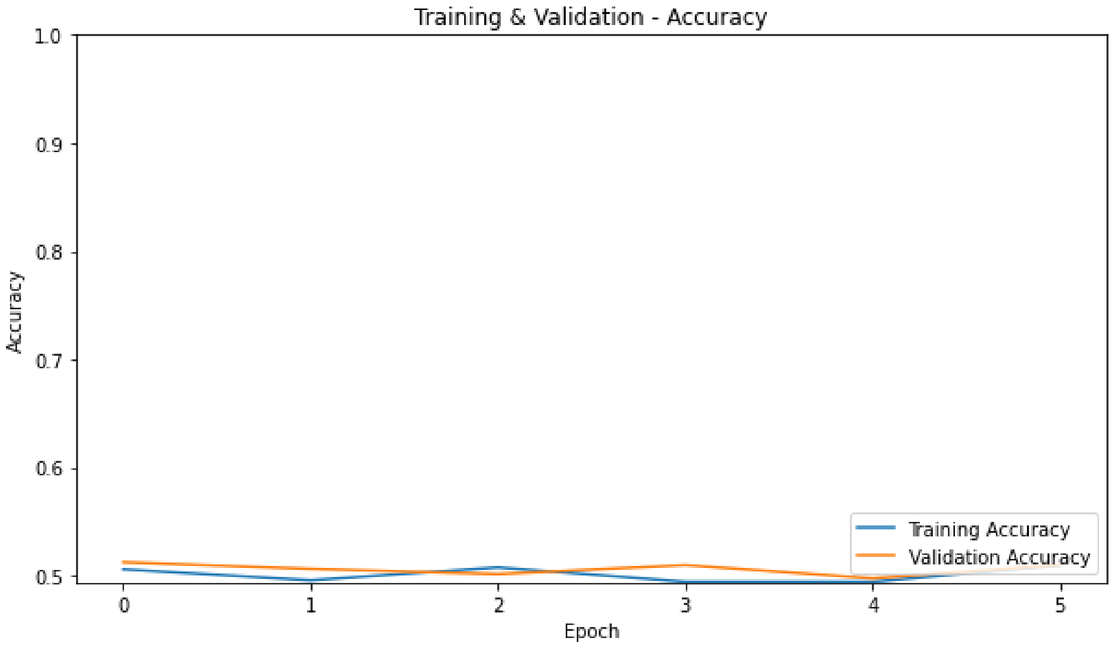
```
MobileNetV2
learning rate 0.01
Downloading MobileNetV2
Epoch 1/10
71/71 [==============================] - 70s 972ms/step - loss: 105.8725 -
accuracy: 0.4547 - val_loss: 68.2724 - val_accuracy: 0.4798

Epoch 00001: val_loss improved from inf to 68.27243, saving model to
models/MobileNetV2exp1.h5
Epoch 2/10
71/71 [==============================] - 68s 963ms/step - loss: 83.2696 -
accuracy: 0.4789 - val_loss: 99.9433 - val_accuracy: 0.5572

Epoch 00002: val_loss did not improve from 68.27243
Epoch 3/10
71/71 [==============================] - 68s 957ms/step - loss: 94.9701 -
accuracy: 0.4749 - val_loss: 99.9873 - val_accuracy: 0.6105

Epoch 00003: val_loss did not improve from 68.27243
Epoch 4/10
71/71 [==============================] - 68s 961ms/step - loss: 68.6673 -
accuracy: 0.5114 - val_loss: 52.8364 - val_accuracy: 0.4859

Epoch 00004: val_loss improved from 68.27243 to 52.83637, saving model to
models/MobileNetV2exp1.h5
Epoch 5/10
71/71 [==============================] - 68s 957ms/step - loss: 55.0493 -
accuracy: 0.4956 - val_loss: 50.9356 - val_accuracy: 0.4674

Epoch 00005: val_loss improved from 52.83637 to 50.93560, saving model to
models/MobileNetV2exp1.h5
Epoch 6/10
71/71 [==============================] - 68s 958ms/step - loss: 49.1239 -
accuracy: 0.5040 - val_loss: 49.7527 - val_accuracy: 0.5396
```

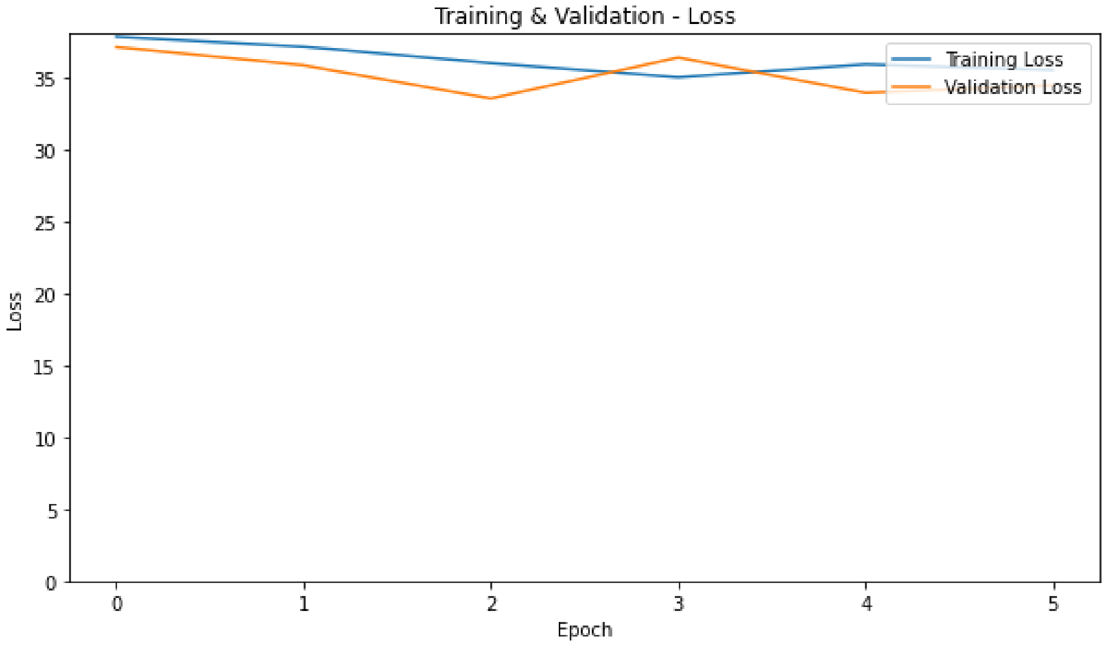
```
Epoch 00006: val_loss improved from 50.93560 to 49.75269, saving model to
models/MobileNetV2exp1.h5
Epoch 7/10
71/71 [==============================] - 68s 957ms/step - loss: 45.3111 -
accuracy: 0.4960 - val_loss: 42.8290 - val_accuracy: 0.5462

Epoch 00007: val_loss improved from 49.75269 to 42.82898, saving model to
models/MobileNetV2exp1.h5
Epoch 8/10
71/71 [==============================] - 68s 957ms/step - loss: 42.3148 -
accuracy: 0.5075 - val_loss: 42.3431 - val_accuracy: 0.5775

Epoch 00008: val_loss improved from 42.82898 to 42.34308, saving model to
models/MobileNetV2exp1.h5
Epoch 9/10
71/71 [==============================] - 68s 961ms/step - loss: 39.6483 -
accuracy: 0.5119 - val_loss: 40.6275 - val_accuracy: 0.5163

Epoch 00009: val_loss improved from 42.34308 to 40.62748, saving model to
models/MobileNetV2exp1.h5
Epoch 10/10
71/71 [==============================] - 68s 958ms/step - loss: 40.6073 -
accuracy: 0.4855 - val_loss: 38.0007 - val_accuracy: 0.4767

Epoch 00010: val_loss improved from 40.62748 to 38.00071, saving model to
models/MobileNetV2exp1.h5
MobileNetV2 Accuracy and Loss plots
```



```
Fine-Tuned MobileNetV2 Training and Validation:
Epoch 1/10
71/71 [==============================] - 70s 973ms/step - loss: 37.8938 -
accuracy: 0.5057 - val_loss: 37.1839 - val_accuracy: 0.5123
```

```
Epoch 00001: val_loss improved from inf to 37.18386, saving model to
models/my_model.h5
Epoch 2/10
71/71 [==============================] - 68s 958ms/step - loss: 37.2070 -
accuracy: 0.4956 - val_loss: 35.9130 - val_accuracy: 0.5062

Epoch 00002: val_loss improved from 37.18386 to 35.91302, saving model to
models/my_model.h5
Epoch 3/10
71/71 [==============================] - 68s 959ms/step - loss: 36.0608 -
accuracy: 0.5075 - val_loss: 33.5982 - val_accuracy: 0.5013

Epoch 00003: val_loss improved from 35.91302 to 33.59819, saving model to
models/my_model.h5
Epoch 4/10
71/71 [==============================] - 68s 960ms/step - loss: 35.0807 -
accuracy: 0.4943 - val_loss: 36.4470 - val_accuracy: 0.5097

Epoch 00004: val_loss did not improve from 33.59819
Epoch 5/10
71/71 [==============================] - 68s 960ms/step - loss: 35.9762 -
accuracy: 0.4943 - val_loss: 33.9969 - val_accuracy: 0.4974

Epoch 00005: val_loss did not improve from 33.59819
Epoch 6/10
71/71 [==============================] - 68s 958ms/step - loss: 35.5847 -
accuracy: 0.5092 - val_loss: 34.5065 - val_accuracy: 0.5097

Epoch 00006: val_loss did not improve from 33.59819
Restoring model weights from the end of the best epoch.
Epoch 00006: early stopping
Fine-Tuned MobileNetV2 Accuracy and Loss plots
```





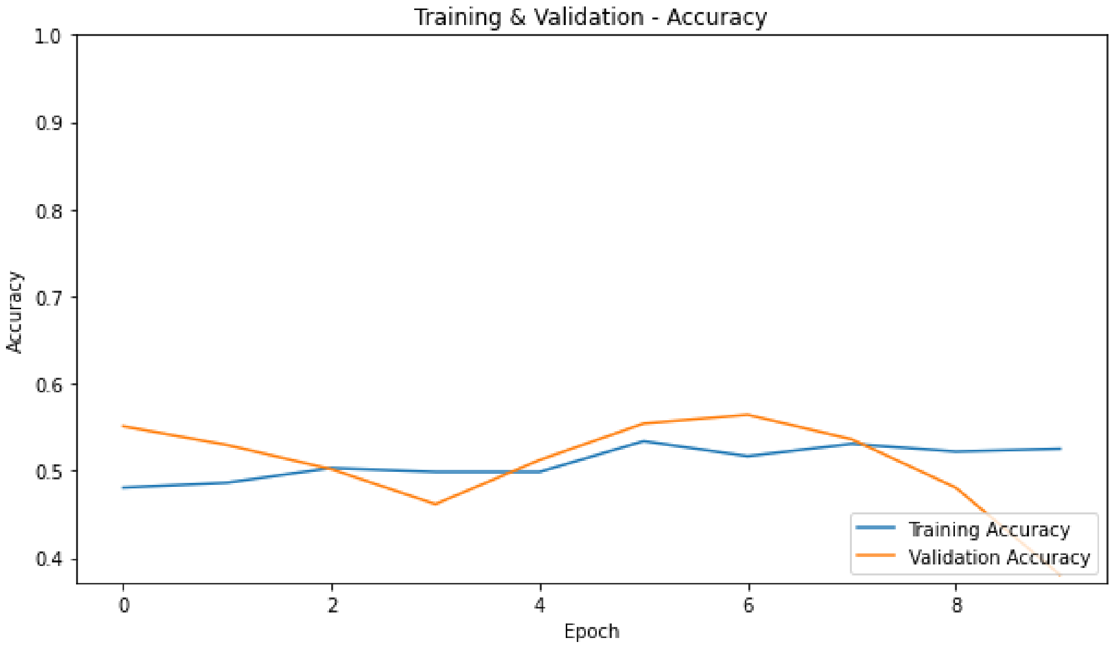
1.6.3 VGG16

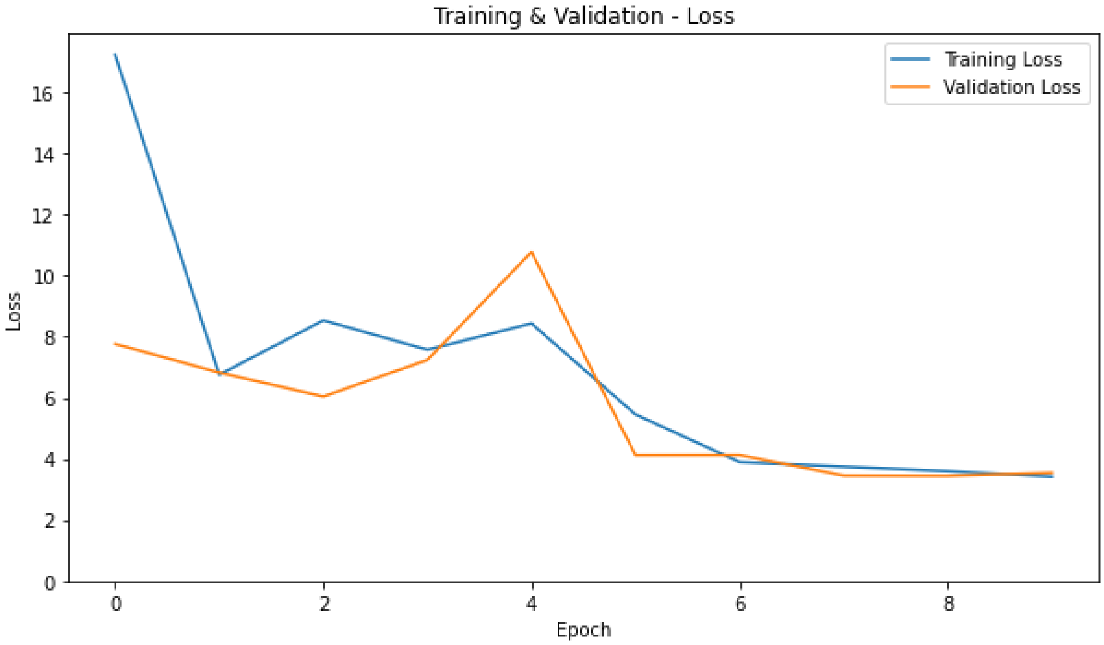
[11]:

```
model_name = PRETRAINED_MODELS[2]
save_model_filepath = 'models/'+ model_name + 'exp1.h5'
logs_dir = 'logs/fit/VGG16exp1'
model = train_and_validate_model(model_name = model_name,
                                 train_generator=train_generator,
                                 valid_generator=valid_generator,
                                 save_model_filepath=save_model_filepath,
                                 logs_dir=logs_dir)
```

```
VGG16
learning rate 0.01
Downloading VGG16
Epoch 1/10
71/71 [==============================] - 71s 962ms/step - loss: 17.2341 -
accuracy: 0.4806 - val_loss: 7.7652 - val_accuracy: 0.5511

Epoch 00001: val_loss improved from inf to 7.76515, saving model to
models/VGG16exp1.h5
Epoch 2/10
71/71 [==============================] - 68s 957ms/step - loss: 6.7573 -
accuracy: 0.4864 - val_loss: 6.8273 - val_accuracy: 0.5295

Epoch 00002: val_loss improved from 7.76515 to 6.82731, saving model to
models/VGG16exp1.h5
Epoch 3/10
71/71 [==============================] - 68s 963ms/step - loss: 8.5359 -
accuracy: 0.5031 - val_loss: 6.0444 - val_accuracy: 0.5018

Epoch 00003: val_loss improved from 6.82731 to 6.04445, saving model to
models/VGG16exp1.h5
Epoch 4/10
71/71 [==============================] - 68s 962ms/step - loss: 7.5747 -
accuracy: 0.4987 - val_loss: 7.2413 - val_accuracy: 0.4617

Epoch 00004: val_loss did not improve from 6.04445
Epoch 5/10
71/71 [==============================] - 68s 958ms/step - loss: 8.4334 -
accuracy: 0.4987 - val_loss: 10.7738 - val_accuracy: 0.5123

Epoch 00005: val_loss did not improve from 6.04445
Epoch 6/10
71/71 [==============================] - 68s 957ms/step - loss: 5.4515 -
accuracy: 0.5339 - val_loss: 4.1194 - val_accuracy: 0.5541

Epoch 00006: val_loss improved from 6.04445 to 4.11941, saving model to
models/VGG16exp1.h5
Epoch 7/10
71/71 [==============================] - 68s 957ms/step - loss: 3.8943 -
accuracy: 0.5167 - val_loss: 4.1227 - val_accuracy: 0.5643

Epoch 00007: val_loss did not improve from 4.11941
Epoch 8/10
```

```
71/71 [==============================] - 68s 961ms/step - loss: 3.7425 -
accuracy: 0.5308 - val_loss: 3.4529 - val_accuracy: 0.5361

Epoch 00008: val_loss improved from 4.11941 to 3.45293, saving model to
models/VGG16exp1.h5
Epoch 9/10
71/71 [==============================] - 68s 957ms/step - loss: 3.5974 -
accuracy: 0.5220 - val_loss: 3.4449 - val_accuracy: 0.4806

Epoch 00009: val_loss improved from 3.45293 to 3.44486, saving model to
models/VGG16exp1.h5
Epoch 10/10
71/71 [==============================] - 68s 960ms/step - loss: 3.4222 -
accuracy: 0.5251 - val_loss: 3.5436 - val_accuracy: 0.3807

Epoch 00010: val_loss did not improve from 3.44486
VGG16 Accuracy and Loss plots
```

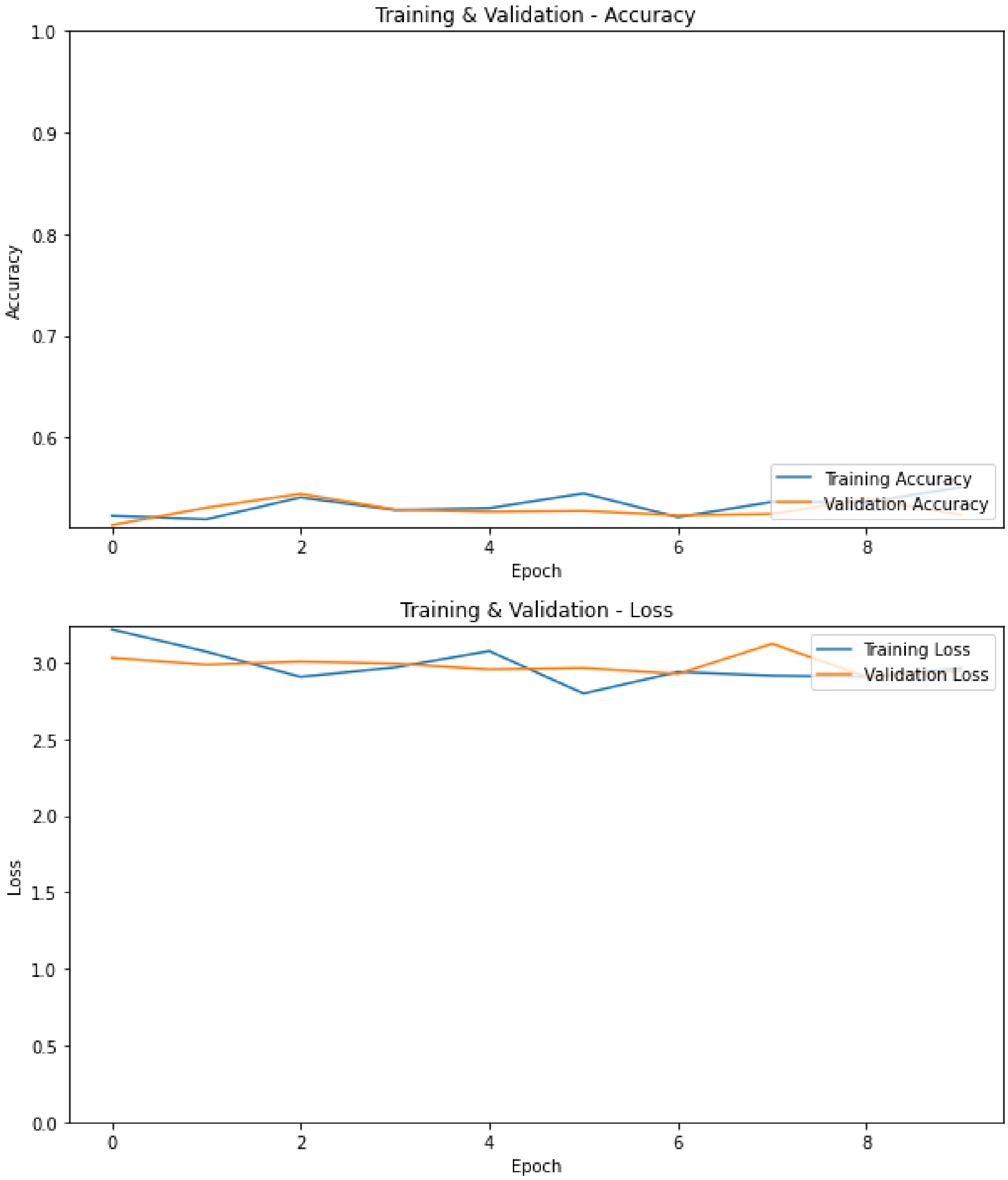




```
Fine-Tuned VGG16 Training and Validation:
Epoch 1/10
71/71 [==============================] - 69s 971ms/step - loss: 3.2178 -
accuracy: 0.5229 - val_loss: 3.0323 - val_accuracy: 0.5136
```

```
Epoch 00001: val_loss improved from inf to 3.03227, saving model to
models/my_model.h5
Epoch 2/10
71/71 [==============================] - 68s 960ms/step - loss: 3.0726 -
accuracy: 0.5194 - val_loss: 2.9876 - val_accuracy: 0.5308

Epoch 00002: val_loss improved from 3.03227 to 2.98765, saving model to
models/my_model.h5
Epoch 3/10
71/71 [==============================] - 68s 958ms/step - loss: 2.9079 -
accuracy: 0.5409 - val_loss: 3.0077 - val_accuracy: 0.5445

Epoch 00003: val_loss did not improve from 2.98765
Epoch 4/10
71/71 [==============================] - 68s 958ms/step - loss: 2.9693 -
accuracy: 0.5286 - val_loss: 2.9946 - val_accuracy: 0.5290

Epoch 00004: val_loss did not improve from 2.98765
Epoch 5/10
71/71 [==============================] - 68s 962ms/step - loss: 3.0764 -
accuracy: 0.5304 - val_loss: 2.9569 - val_accuracy: 0.5268

Epoch 00005: val_loss improved from 2.98765 to 2.95689, saving model to
models/my_model.h5
Epoch 6/10
71/71 [==============================] - 67s 953ms/step - loss: 2.7987 -
accuracy: 0.5449 - val_loss: 2.9659 - val_accuracy: 0.5277

Epoch 00006: val_loss did not improve from 2.95689
Epoch 7/10
71/71 [==============================] - 67s 956ms/step - loss: 2.9401 -
accuracy: 0.5216 - val_loss: 2.9276 - val_accuracy: 0.5233

Epoch 00007: val_loss improved from 2.95689 to 2.92760, saving model to
models/my_model.h5
Epoch 8/10
71/71 [==============================] - 68s 958ms/step - loss: 2.9153 -
accuracy: 0.5365 - val_loss: 3.1242 - val_accuracy: 0.5246

Epoch 00008: val_loss did not improve from 2.92760
Epoch 9/10
71/71 [==============================] - 68s 957ms/step - loss: 2.9076 -
accuracy: 0.5361 - val_loss: 2.9075 - val_accuracy: 0.5396

Epoch 00009: val_loss improved from 2.92760 to 2.90750, saving model to
models/my_model.h5
Epoch 10/10
```

```
71/71 [==============================] - 68s 962ms/step - loss: 2.9663 -
accuracy: 0.5511 - val_loss: 2.9462 - val_accuracy: 0.5238

Epoch 00010: val_loss did not improve from 2.90750
Fine-Tuned VGG16 Accuracy and Loss plots
```



1.7 Experiment 2: Balance the dataset

[12]:

```
nih_xrays_df = pd.read_csv('data/processed/prepared_data_entry_2017.csv')
```

[13]:

```
def sample_with_weights(df, all_labels, num_samples: int = 40000):
    for lbl in all_labels:
        df[lbl] = df['finding_label'].map(lambda find: 1 if lbl in find else 0)
    df['encoding'] = [[1 if l in lbl.split('|') else 0 for l in all_labels] for␣
↪lbl in nih_xrays_df['finding_label']]

    class_count = {}
    for lbl in all_labels:
        class_count[lbl] = df[lbl].sum()

    classweight = {}
    for lbl in all_labels :
        classweight[lbl] = 1/class_count[lbl]

    classweight['NoFinding'] /= 2   #Extra penalising the none class
    def apply_weights(row):
        weight = 0
        for lbl in all_labels:
            if(row[lbl]==1):
                weight += classweight[lbl]
        return weight
    new_weights = df.apply(apply_weights, axis=1)
    sampled_data = df.sample(50000, weights = new_weights)


    nih_required_columns = {
            'patient_id',
            'image_name',
            'path',
            'finding_label'
        }


    sampled_data = sampled_data[nih_required_columns]


    group_shuffle_split = GroupShuffleSplit(n_splits=1, train_size=0.8,␣
↪random_state=42)

    for train_idx, valid_idx in group_shuffle_split.split(sampled_data[:None],\
        groups=sampled_data[:None]['patient_id'].values):
        train_df = sampled_data.iloc[train_idx]
        valid_df = sampled_data.iloc[valid_idx]
    return train_df, valid_df
```

[14]:

```
train_df, valid_df =␣
 ↪sample_with_weights(nih_xrays_df,all_labels,num_samples=40000)
sampled_train_gen = train_model.
 ↪get_image_data_generator(train_df,batch_size=BATCH_SIZE,image_size=IMAGE_SIZE,lables=all_la
sampled_valid_gen = train_model.
 ↪get_image_data_generator(valid_df,batch_size=BATCH_SIZE,image_size=IMAGE_SIZE,lables=all_la
```

```
Found 23993 validated image filenames belonging to 15 classes.
Found 6020 validated image filenames belonging to 15 classes.
```

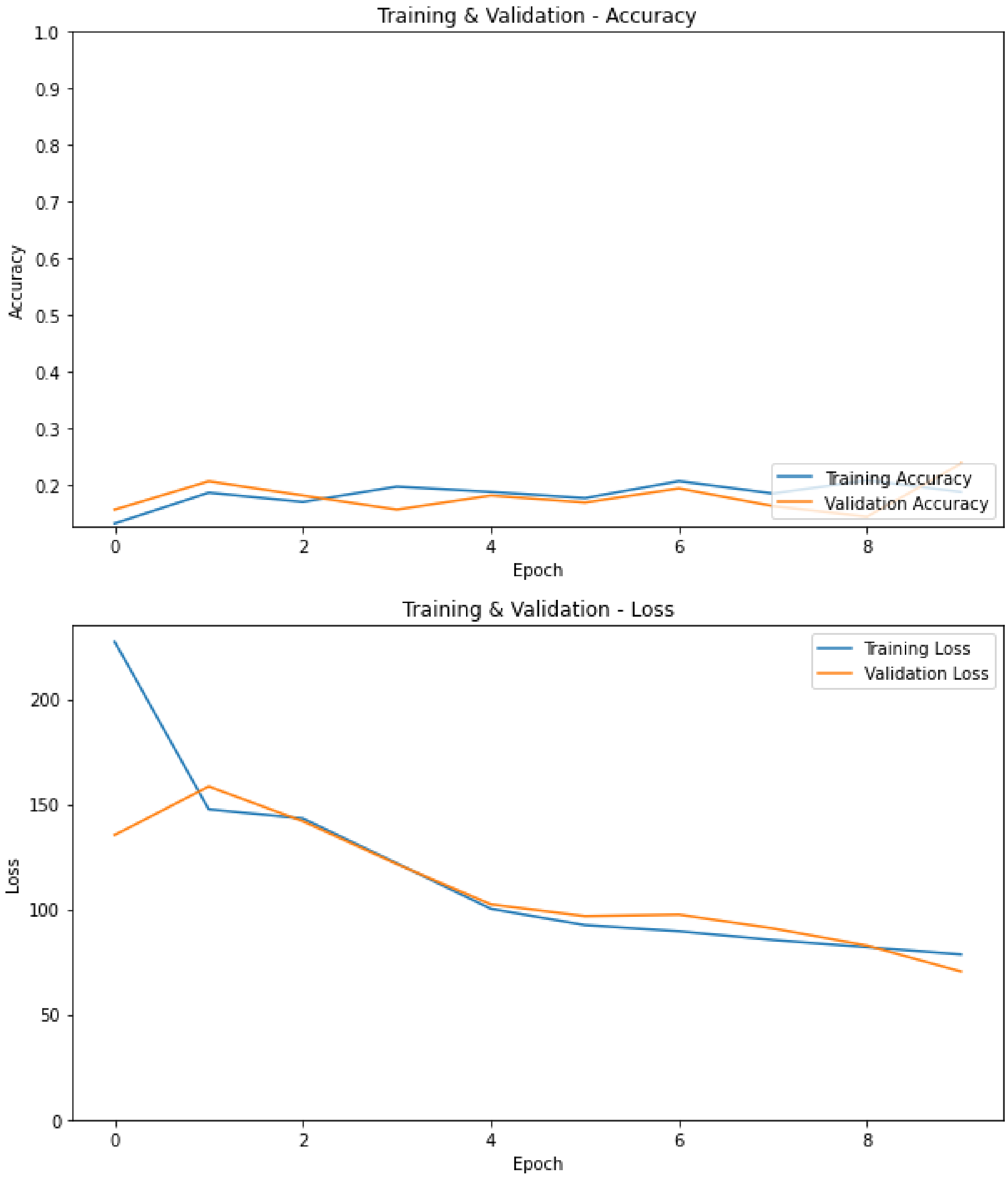
1.7.1 ResNet50V2

[15]:

```
model_name = PRETRAINED_MODELS[0]
save_model_filepath = 'models/'+ model_name + 'exp2.h5'
logs_dir = 'logs/fit/ResNet50V2exp2'
model = train_and_validate_model(model_name = model_name,
                                 train_generator=sampled_train_gen,
                                 valid_generator=sampled_valid_gen,
                                 save_model_filepath=save_model_filepath,
                                 logs_dir=logs_dir)
```

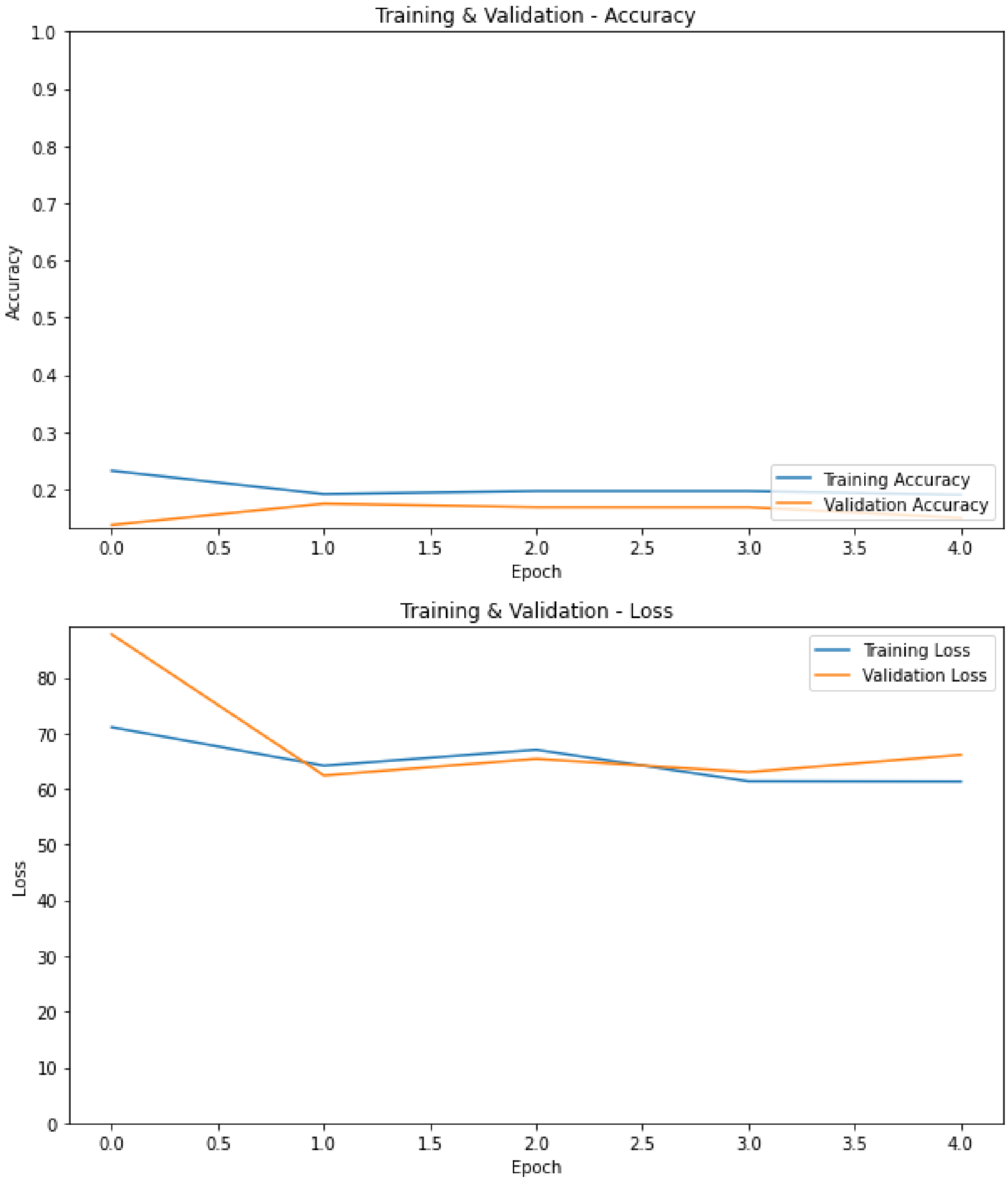
```
ResNet50V2
learning rate 0.01
Downloading ResNet50V2
Epoch 1/10
23/23 [==============================] - 17s 669ms/step - loss: 227.3232 -
accuracy: 0.1318 - val_loss: 135.4479 - val_accuracy: 0.1562

Epoch 00001: val_loss improved from inf to 135.44791, saving model to
models/ResNet50V2exp2.h5
Epoch 2/10
23/23 [==============================] - 14s 615ms/step - loss: 147.6189 -
accuracy: 0.1861 - val_loss: 158.4441 - val_accuracy: 0.2062

Epoch 00002: val_loss did not improve from 135.44791
Epoch 3/10
23/23 [==============================] - 14s 612ms/step - loss: 143.3174 -
accuracy: 0.1698 - val_loss: 141.9806 - val_accuracy: 0.1813

Epoch 00003: val_loss did not improve from 135.44791
Epoch 4/10
23/23 [==============================] - 14s 606ms/step - loss: 121.9325 -
accuracy: 0.1970 - val_loss: 121.5578 - val_accuracy: 0.1562
```

```
Epoch 00004: val_loss improved from 135.44791 to 121.55784, saving model to
models/ResNet50V2exp2.h5
Epoch 5/10
23/23 [==============================] - 14s 611ms/step - loss: 100.2714 -
accuracy: 0.1875 - val_loss: 102.3794 - val_accuracy: 0.1813

Epoch 00005: val_loss improved from 121.55784 to 102.37938, saving model to
models/ResNet50V2exp2.h5
Epoch 6/10
23/23 [==============================] - 14s 604ms/step - loss: 92.5760 -
accuracy: 0.1766 - val_loss: 96.8073 - val_accuracy: 0.1688

Epoch 00006: val_loss improved from 102.37938 to 96.80727, saving model to
models/ResNet50V2exp2.h5
Epoch 7/10
23/23 [==============================] - 14s 614ms/step - loss: 89.6441 -
accuracy: 0.2065 - val_loss: 97.5495 - val_accuracy: 0.1937

Epoch 00007: val_loss did not improve from 96.80727
Epoch 8/10
23/23 [==============================] - 14s 606ms/step - loss: 85.4766 -
accuracy: 0.1848 - val_loss: 91.0133 - val_accuracy: 0.1625

Epoch 00008: val_loss improved from 96.80727 to 91.01328, saving model to
models/ResNet50V2exp2.h5
Epoch 9/10
23/23 [==============================] - 14s 607ms/step - loss: 82.0316 -
accuracy: 0.2079 - val_loss: 82.8124 - val_accuracy: 0.1437

Epoch 00009: val_loss improved from 91.01328 to 82.81242, saving model to
models/ResNet50V2exp2.h5
Epoch 10/10
23/23 [==============================] - 14s 610ms/step - loss: 78.6213 -
accuracy: 0.1875 - val_loss: 70.4454 - val_accuracy: 0.2375

Epoch 00010: val_loss improved from 82.81242 to 70.44543, saving model to
models/ResNet50V2exp2.h5
ResNet50V2 Accuracy and Loss plots
```



```
Fine-Tuned ResNet50V2 Training and Validation:
Epoch 1/10
23/23 [==============================] - 17s 667ms/step - loss: 71.1223 -
accuracy: 0.2323 - val_loss: 87.8081 - val_accuracy: 0.1375
```

```
Epoch 00001: val_loss improved from inf to 87.80807, saving model to
models/my_model.h5
Epoch 2/10
23/23 [==============================] - 14s 608ms/step - loss: 64.1974 -
accuracy: 0.1916 - val_loss: 62.4603 - val_accuracy: 0.1750

Epoch 00002: val_loss improved from 87.80807 to 62.46031, saving model to
models/my_model.h5
Epoch 3/10
23/23 [==============================] - 14s 613ms/step - loss: 67.0687 -
accuracy: 0.1970 - val_loss: 65.4240 - val_accuracy: 0.1688

Epoch 00003: val_loss did not improve from 62.46031
Epoch 4/10
23/23 [==============================] - 14s 607ms/step - loss: 61.4066 -
accuracy: 0.1970 - val_loss: 63.0498 - val_accuracy: 0.1688

Epoch 00004: val_loss did not improve from 62.46031
Epoch 5/10
23/23 [==============================] - 14s 613ms/step - loss: 61.3567 -
accuracy: 0.1902 - val_loss: 66.1511 - val_accuracy: 0.1500

Epoch 00005: val_loss did not improve from 62.46031
Restoring model weights from the end of the best epoch.
Epoch 00005: early stopping
Fine-Tuned ResNet50V2 Accuracy and Loss plots
```

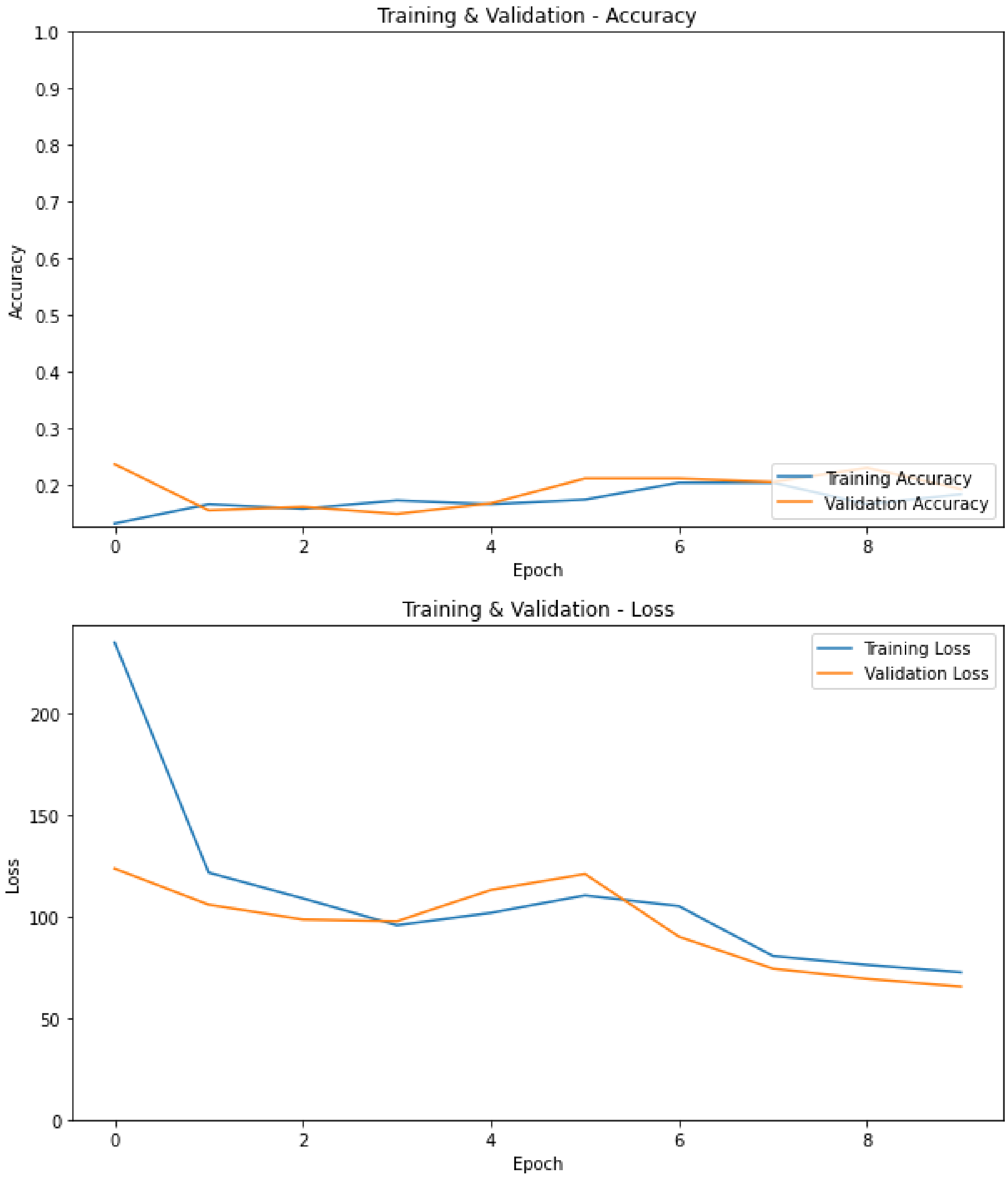


1.7.2 MobileNetV2

[16]:
```
model_name = PRETRAINED_MODELS[1]
save_model_filepath = 'models/'+ model_name + 'exp2.h5'
logs_dir = 'logs/fit/MobileNetV2exp2'
model = train_and_validate_model(model_name = model_name,
                                 train_generator=sampled_train_gen,
                                 valid_generator=sampled_valid_gen,
                                 save_model_filepath=save_model_filepath,
                                 logs_dir=logs_dir)
```

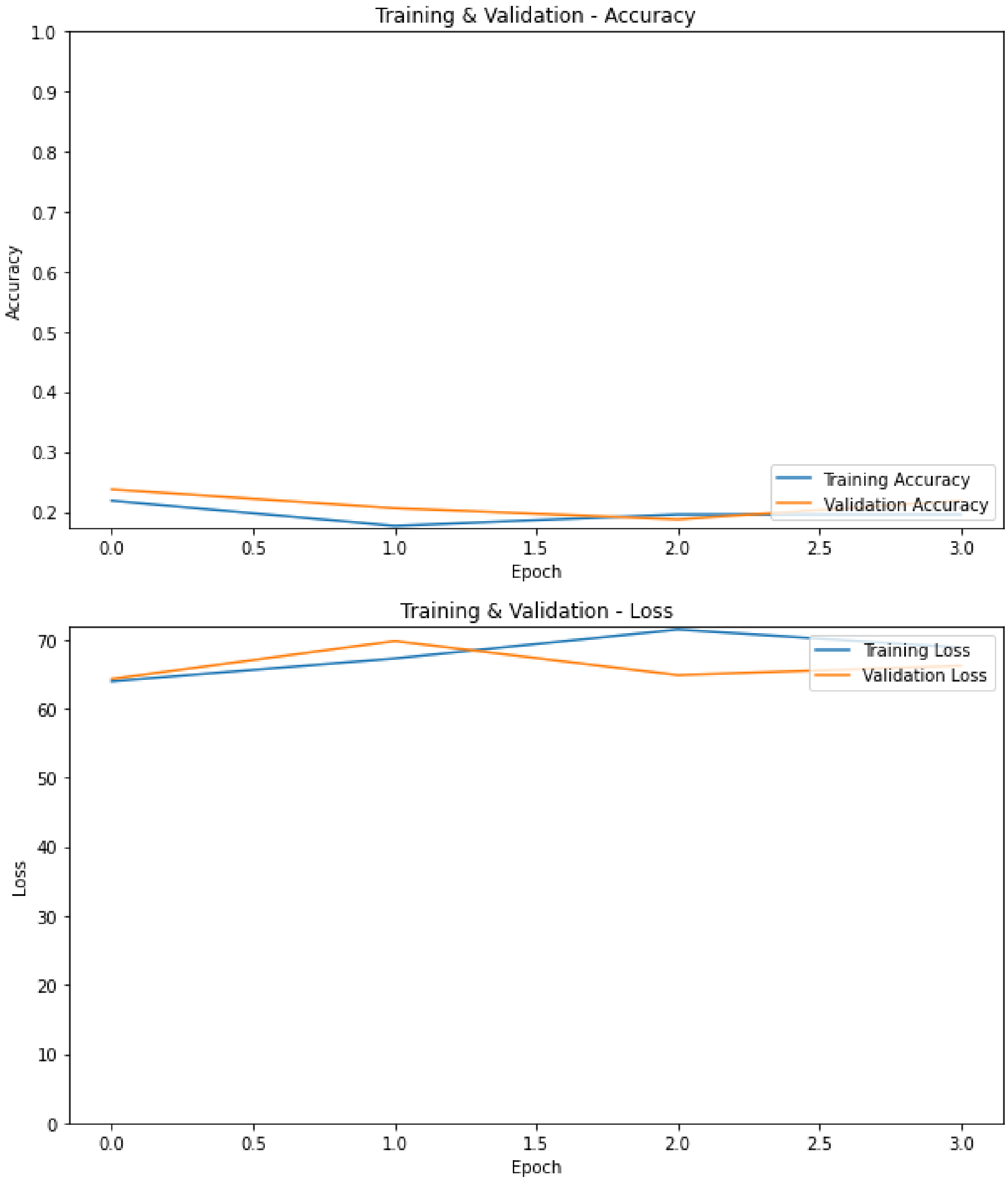
```
MobileNetV2
learning rate 0.01
Downloading MobileNetV2
Epoch 1/10
23/23 [==============================] - 17s 661ms/step - loss: 234.5731 -
accuracy: 0.1332 - val_loss: 123.3951 - val_accuracy: 0.2375

Epoch 00001: val_loss improved from inf to 123.39513, saving model to
models/MobileNetV2exp2.h5
Epoch 2/10
23/23 [==============================] - 14s 609ms/step - loss: 121.4135 -
accuracy: 0.1671 - val_loss: 105.6474 - val_accuracy: 0.1562

Epoch 00002: val_loss improved from 123.39513 to 105.64744, saving model to
models/MobileNetV2exp2.h5
Epoch 3/10
23/23 [==============================] - 14s 603ms/step - loss: 108.7570 -
accuracy: 0.1590 - val_loss: 98.4326 - val_accuracy: 0.1625

Epoch 00003: val_loss improved from 105.64744 to 98.43264, saving model to
models/MobileNetV2exp2.h5
Epoch 4/10
23/23 [==============================] - 14s 613ms/step - loss: 95.6225 -
accuracy: 0.1739 - val_loss: 97.5393 - val_accuracy: 0.1500

Epoch 00004: val_loss improved from 98.43264 to 97.53934, saving model to
models/MobileNetV2exp2.h5
Epoch 5/10
23/23 [==============================] - 14s 607ms/step - loss: 101.6591 -
accuracy: 0.1671 - val_loss: 112.8824 - val_accuracy: 0.1688

Epoch 00005: val_loss did not improve from 97.53934
Epoch 6/10
23/23 [==============================] - 14s 614ms/step - loss: 110.2402 -
accuracy: 0.1753 - val_loss: 120.7626 - val_accuracy: 0.2125

Epoch 00006: val_loss did not improve from 97.53934
Epoch 7/10
23/23 [==============================] - 14s 605ms/step - loss: 104.9442 -
accuracy: 0.2052 - val_loss: 89.9691 - val_accuracy: 0.2125

Epoch 00007: val_loss improved from 97.53934 to 89.96910, saving model to
models/MobileNetV2exp2.h5
```

```
Epoch 8/10
23/23 [==============================] - 14s 604ms/step - loss: 80.4686 -
accuracy: 0.2052 - val_loss: 74.2304 - val_accuracy: 0.2062

Epoch 00008: val_loss improved from 89.96910 to 74.23045, saving model to
models/MobileNetV2exp2.h5
Epoch 9/10
23/23 [==============================] - 14s 611ms/step - loss: 76.0711 -
accuracy: 0.1671 - val_loss: 69.2441 - val_accuracy: 0.2313

Epoch 00009: val_loss improved from 74.23045 to 69.24414, saving model to
models/MobileNetV2exp2.h5
Epoch 10/10
23/23 [==============================] - 14s 605ms/step - loss: 72.4063 -
accuracy: 0.1848 - val_loss: 65.3695 - val_accuracy: 0.1937

Epoch 00010: val_loss improved from 69.24414 to 65.36948, saving model to
models/MobileNetV2exp2.h5
MobileNetV2 Accuracy and Loss plots
```



```
Fine-Tuned MobileNetV2 Training and Validation:
Epoch 1/10
23/23 [==============================] - 16s 658ms/step - loss: 63.9529 -
accuracy: 0.2188 - val_loss: 64.3041 - val_accuracy: 0.2375
```

```
Epoch 00001: val_loss improved from inf to 64.30409, saving model to
models/my_model.h5
Epoch 2/10
23/23 [==============================] - 14s 613ms/step - loss: 67.2773 -
accuracy: 0.1766 - val_loss: 69.8011 - val_accuracy: 0.2062

Epoch 00002: val_loss did not improve from 64.30409
Epoch 3/10
23/23 [==============================] - 14s 611ms/step - loss: 71.4625 -
accuracy: 0.1957 - val_loss: 64.8600 - val_accuracy: 0.1875

Epoch 00003: val_loss did not improve from 64.30409
Epoch 4/10
23/23 [==============================] - 14s 611ms/step - loss: 68.8488 -
accuracy: 0.1957 - val_loss: 66.2205 - val_accuracy: 0.2188

Epoch 00004: val_loss did not improve from 64.30409
Restoring model weights from the end of the best epoch.
Epoch 00004: early stopping
Fine-Tuned MobileNetV2 Accuracy and Loss plots
```

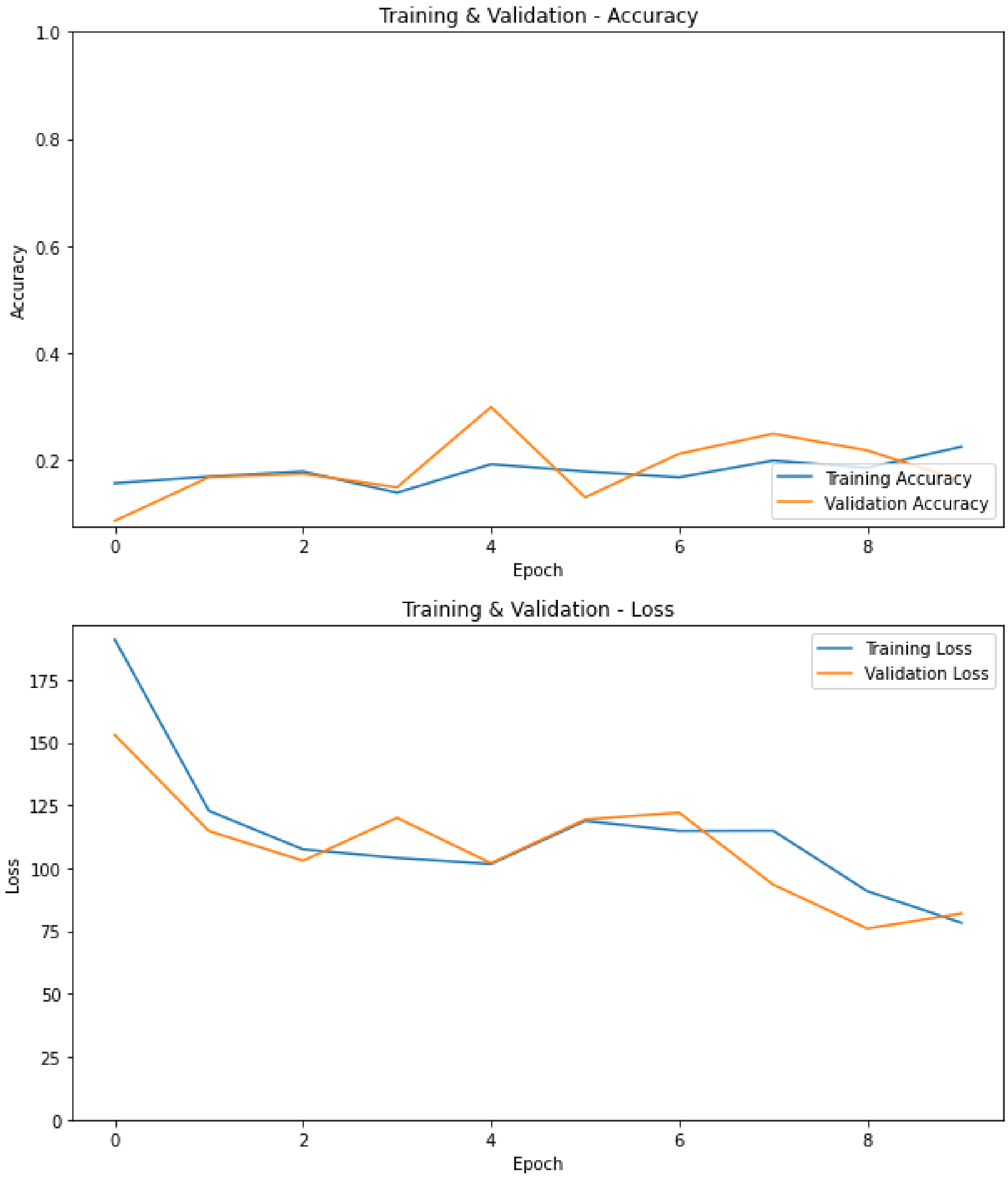


1.7.3 VGG16

[17]:
```
# model_name = PRETRAINED_MODELS[2]
save_model_filepath = 'models/'+ model_name + 'exp2.h5'
logs_dir = 'logs/fit/VGG16exp2'
model = train_and_validate_model(model_name = model_name,
                                 train_generator=sampled_train_gen,
                                 valid_generator=sampled_valid_gen,
                                 save_model_filepath=save_model_filepath,
                                 logs_dir=logs_dir)
```

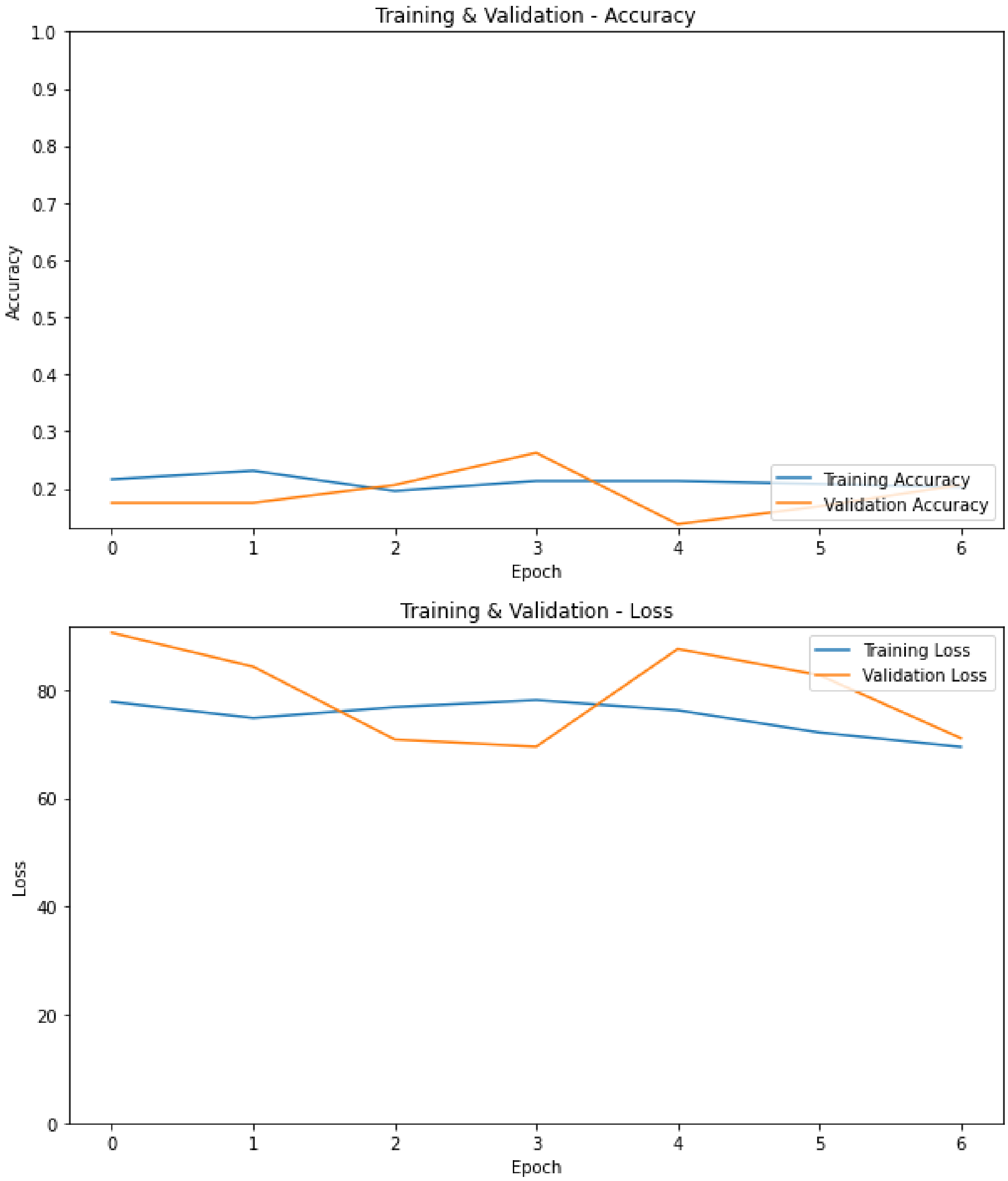
```
MobileNetV2
learning rate 0.01
Downloading MobileNetV2
Epoch 1/10
23/23 [==============================] - 17s 680ms/step - loss: 190.8648 -
accuracy: 0.1576 - val_loss: 152.9774 - val_accuracy: 0.0875

Epoch 00001: val_loss improved from inf to 152.97742, saving model to
models/MobileNetV2exp2.h5
Epoch 2/10
23/23 [==============================] - 14s 610ms/step - loss: 122.8138 -
accuracy: 0.1698 - val_loss: 114.7966 - val_accuracy: 0.1688

Epoch 00002: val_loss improved from 152.97742 to 114.79661, saving model to
models/MobileNetV2exp2.h5
Epoch 3/10
23/23 [==============================] - 14s 608ms/step - loss: 107.4997 -
accuracy: 0.1793 - val_loss: 102.9627 - val_accuracy: 0.1750

Epoch 00003: val_loss improved from 114.79661 to 102.96275, saving model to
models/MobileNetV2exp2.h5
Epoch 4/10
23/23 [==============================] - 14s 608ms/step - loss: 104.0811 -
accuracy: 0.1399 - val_loss: 120.0657 - val_accuracy: 0.1500

Epoch 00004: val_loss did not improve from 102.96275
Epoch 5/10
23/23 [==============================] - 14s 610ms/step - loss: 101.6908 -
accuracy: 0.1929 - val_loss: 102.0034 - val_accuracy: 0.3000

Epoch 00005: val_loss improved from 102.96275 to 102.00336, saving model to
models/MobileNetV2exp2.h5
Epoch 6/10
23/23 [==============================] - 14s 604ms/step - loss: 118.8334 -
accuracy: 0.1793 - val_loss: 119.4135 - val_accuracy: 0.1312

Epoch 00006: val_loss did not improve from 102.00336
Epoch 7/10
23/23 [==============================] - 14s 615ms/step - loss: 114.8279 -
accuracy: 0.1685 - val_loss: 122.1105 - val_accuracy: 0.2125

Epoch 00007: val_loss did not improve from 102.00336
Epoch 8/10
```

```
23/23 [==============================] - 14s 606ms/step - loss: 114.9141 -
accuracy: 0.1997 - val_loss: 93.4615 - val_accuracy: 0.2500

Epoch 00008: val_loss improved from 102.00336 to 93.46152, saving model to
models/MobileNetV2exp2.h5
Epoch 9/10
23/23 [==============================] - 14s 614ms/step - loss: 90.8407 -
accuracy: 0.1861 - val_loss: 76.0203 - val_accuracy: 0.2188

Epoch 00009: val_loss improved from 93.46152 to 76.02030, saving model to
models/MobileNetV2exp2.h5
Epoch 10/10
23/23 [==============================] - 14s 606ms/step - loss: 78.3126 -
accuracy: 0.2255 - val_loss: 81.9941 - val_accuracy: 0.1625

Epoch 00010: val_loss did not improve from 76.02030
MobileNetV2 Accuracy and Loss plots
```



```
Fine-Tuned MobileNetV2 Training and Validation:
Epoch 1/10
23/23 [==============================] - 16s 665ms/step - loss: 77.8292 -
accuracy: 0.2160 - val_loss: 90.5927 - val_accuracy: 0.1750
```

```
Epoch 00001: val_loss improved from inf to 90.59265, saving model to
models/my_model.h5
Epoch 2/10
23/23 [==============================] - 14s 607ms/step - loss: 74.8171 -
accuracy: 0.2310 - val_loss: 84.3193 - val_accuracy: 0.1750

Epoch 00002: val_loss improved from 90.59265 to 84.31928, saving model to
models/my_model.h5
Epoch 3/10
23/23 [==============================] - 14s 608ms/step - loss: 76.8379 -
accuracy: 0.1957 - val_loss: 70.8639 - val_accuracy: 0.2062

Epoch 00003: val_loss improved from 84.31928 to 70.86387, saving model to
models/my_model.h5
Epoch 4/10
23/23 [==============================] - 14s 608ms/step - loss: 78.1323 -
accuracy: 0.2133 - val_loss: 69.5221 - val_accuracy: 0.2625

Epoch 00004: val_loss improved from 70.86387 to 69.52206, saving model to
models/my_model.h5
Epoch 5/10
23/23 [==============================] - 14s 607ms/step - loss: 76.2506 -
accuracy: 0.2133 - val_loss: 87.5804 - val_accuracy: 0.1375

Epoch 00005: val_loss did not improve from 69.52206
Epoch 6/10
23/23 [==============================] - 14s 610ms/step - loss: 72.1555 -
accuracy: 0.2079 - val_loss: 82.7526 - val_accuracy: 0.1688

Epoch 00006: val_loss did not improve from 69.52206
Epoch 7/10
23/23 [==============================] - 14s 615ms/step - loss: 69.4824 -
accuracy: 0.2024 - val_loss: 71.1104 - val_accuracy: 0.2062

Epoch 00007: val_loss did not improve from 69.52206
Restoring model weights from the end of the best epoch.
Epoch 00007: early stopping
Fine-Tuned MobileNetV2 Accuracy and Loss plots
```



1.7.4 Experiment 3: Sub Sampling Classes

[18]:
```
sub_samples = ['Cardiomegaly','Effusion','Emphysema', 'Fibrosis',␣
↪'Infiltration', 'Pneumonia', 'Pneumothorax','Pleural_Thickening']
```

```
[19]: sub_nih_xrays_train_df  = nih_xrays_train_df␣
       ↪[nih_xrays_train_df['finding_label'].isin(sub_samples)]
      sub_nih_xrays_valid_df = nih_xrays_valid_df[nih_xrays_valid_df['finding_label'].
       ↪isin(sub_samples)]
      sub_sampled_train_gen = train_model.
       ↪get_image_data_generator(sub_nih_xrays_train_df,batch_size=BATCH_SIZE,image_size=IMAGE_SIZE
      sub_sampled_valid_gen = train_model.
       ↪get_image_data_generator(sub_nih_xrays_valid_df,batch_size=BATCH_SIZE,image_size=IMAGE_SIZE
```

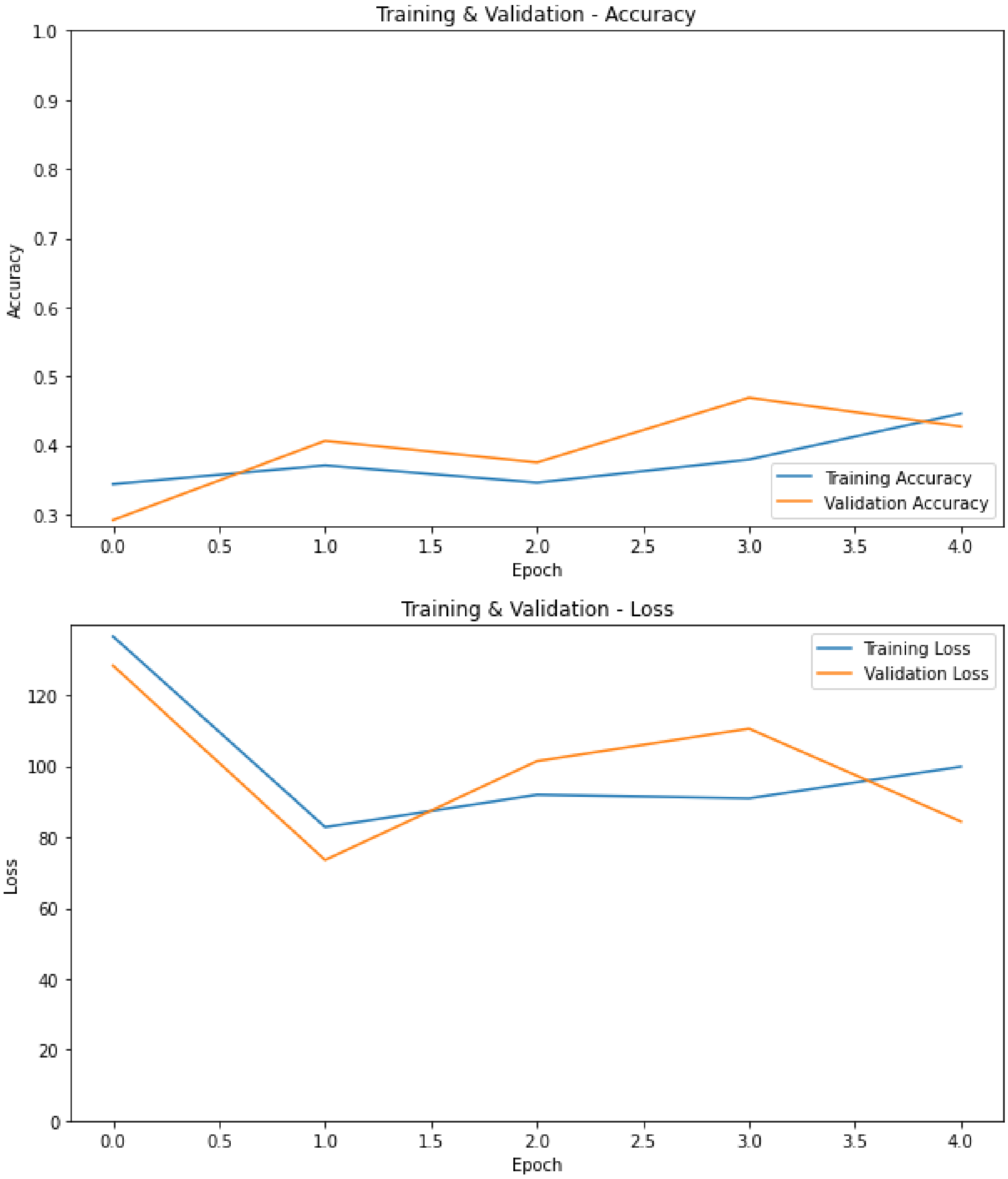
```
Found 15794 validated image filenames belonging to 8 classes.
Found 4060 validated image filenames belonging to 8 classes.
```

1.7.5 ResNet50V2

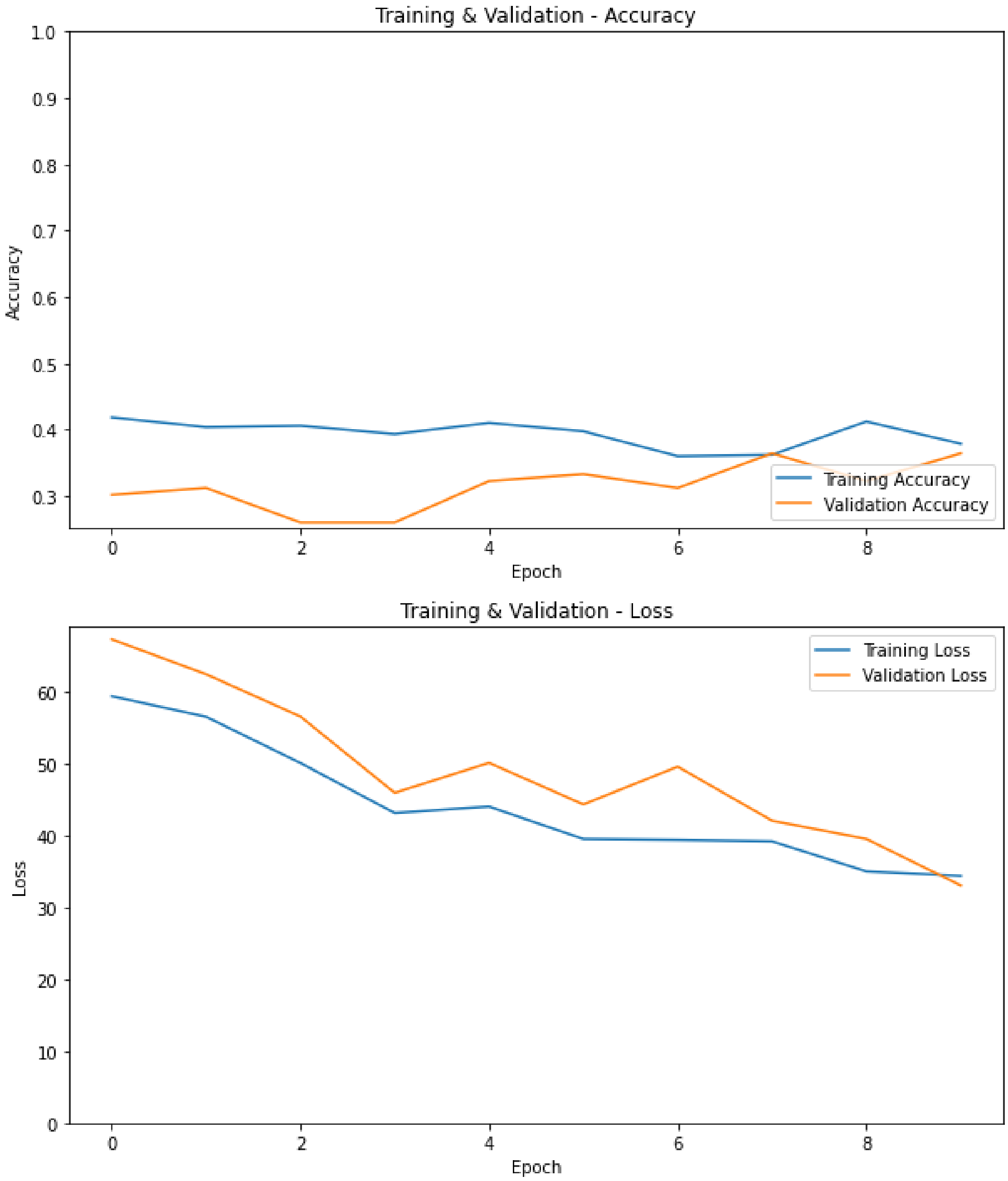
```
[20]: model_name = PRETRAINED_MODELS[0]
      save_model_filepath = 'models/'+ model_name + 'exp3.h5'
      logs_dir = 'logs/fit/ResNet50V2exp3'
      model = train_and_validate_model(model_name = model_name,
                                       train_generator=sub_sampled_train_gen,
                                       valid_generator=sub_sampled_valid_gen,
                                       save_model_filepath=save_model_filepath,
                                       logs_dir=logs_dir,
                                       num_classes=len(sub_samples))
```

```
ResNet50V2
learning rate 0.01
Downloading ResNet50V2
Epoch 1/10
15/15 [==============================] - 12s 702ms/step - loss: 136.6870 -
accuracy: 0.3438 - val_loss: 128.4361 - val_accuracy: 0.2917

Epoch 00001: val_loss improved from inf to 128.43614, saving model to
models/ResNet50V2exp3.h5
Epoch 2/10
15/15 [==============================] - 9s 607ms/step - loss: 82.8745 -
accuracy: 0.3708 - val_loss: 73.5483 - val_accuracy: 0.4062

Epoch 00002: val_loss improved from 128.43614 to 73.54834, saving model to
models/ResNet50V2exp3.h5
Epoch 3/10
15/15 [==============================] - 9s 619ms/step - loss: 91.9664 -
accuracy: 0.3458 - val_loss: 101.4882 - val_accuracy: 0.3750

Epoch 00003: val_loss did not improve from 73.54834
Epoch 4/10
15/15 [==============================] - 9s 608ms/step - loss: 90.9545 -
accuracy: 0.3792 - val_loss: 110.6706 - val_accuracy: 0.4688
```

```
Epoch 00004: val_loss did not improve from 73.54834
Epoch 5/10
15/15 [==============================] - 9s 607ms/step - loss: 99.9130 -
accuracy: 0.4458 - val_loss: 84.4380 - val_accuracy: 0.4271

Epoch 00005: val_loss did not improve from 73.54834
Restoring model weights from the end of the best epoch.
Epoch 00005: early stopping
ResNet50V2 Accuracy and Loss plots
```



```
Fine-Tuned ResNet50V2 Training and Validation:
Epoch 1/10
15/15 [==============================] - 12s 696ms/step - loss: 59.3330 -
accuracy: 0.4187 - val_loss: 67.2428 - val_accuracy: 0.3021

Epoch 00001: val_loss improved from inf to 67.24281, saving model to
models/my_model.h5
Epoch 2/10
15/15 [==============================] - 9s 612ms/step - loss: 56.4729 -
accuracy: 0.4042 - val_loss: 62.3849 - val_accuracy: 0.3125

Epoch 00002: val_loss improved from 67.24281 to 62.38486, saving model to
models/my_model.h5
Epoch 3/10
15/15 [==============================] - 9s 614ms/step - loss: 50.0476 -
accuracy: 0.4062 - val_loss: 56.5075 - val_accuracy: 0.2604

Epoch 00003: val_loss improved from 62.38486 to 56.50751, saving model to
models/my_model.h5
Epoch 4/10
15/15 [==============================] - 9s 616ms/step - loss: 43.0891 -
accuracy: 0.3938 - val_loss: 45.8979 - val_accuracy: 0.2604

Epoch 00004: val_loss improved from 56.50751 to 45.89791, saving model to
models/my_model.h5
Epoch 5/10
15/15 [==============================] - 9s 610ms/step - loss: 43.9806 -
accuracy: 0.4104 - val_loss: 50.0645 - val_accuracy: 0.3229

Epoch 00005: val_loss did not improve from 45.89791
Epoch 6/10
15/15 [==============================] - 9s 617ms/step - loss: 39.4737 -
accuracy: 0.3979 - val_loss: 44.2855 - val_accuracy: 0.3333

Epoch 00006: val_loss improved from 45.89791 to 44.28554, saving model to
models/my_model.h5
Epoch 7/10
15/15 [==============================] - 9s 607ms/step - loss: 39.3460 -
accuracy: 0.3604 - val_loss: 49.5343 - val_accuracy: 0.3125

Epoch 00007: val_loss did not improve from 44.28554
Epoch 8/10
15/15 [==============================] - 9s 615ms/step - loss: 39.1489 -
accuracy: 0.3625 - val_loss: 42.0220 - val_accuracy: 0.3646

Epoch 00008: val_loss improved from 44.28554 to 42.02197, saving model to
models/my_model.h5
Epoch 9/10
15/15 [==============================] - 9s 613ms/step - loss: 34.9760 -
accuracy: 0.4125 - val_loss: 39.4784 - val_accuracy: 0.3229

Epoch 00009: val_loss improved from 42.02197 to 39.47840, saving model to
models/my_model.h5
Epoch 10/10
15/15 [==============================] - 9s 612ms/step - loss: 34.3505 -
accuracy: 0.3792 - val_loss: 33.0318 - val_accuracy: 0.3646

Epoch 00010: val_loss improved from 39.47840 to 33.03177, saving model to
models/my_model.h5
Fine-Tuned ResNet50V2 Accuracy and Loss plots
```

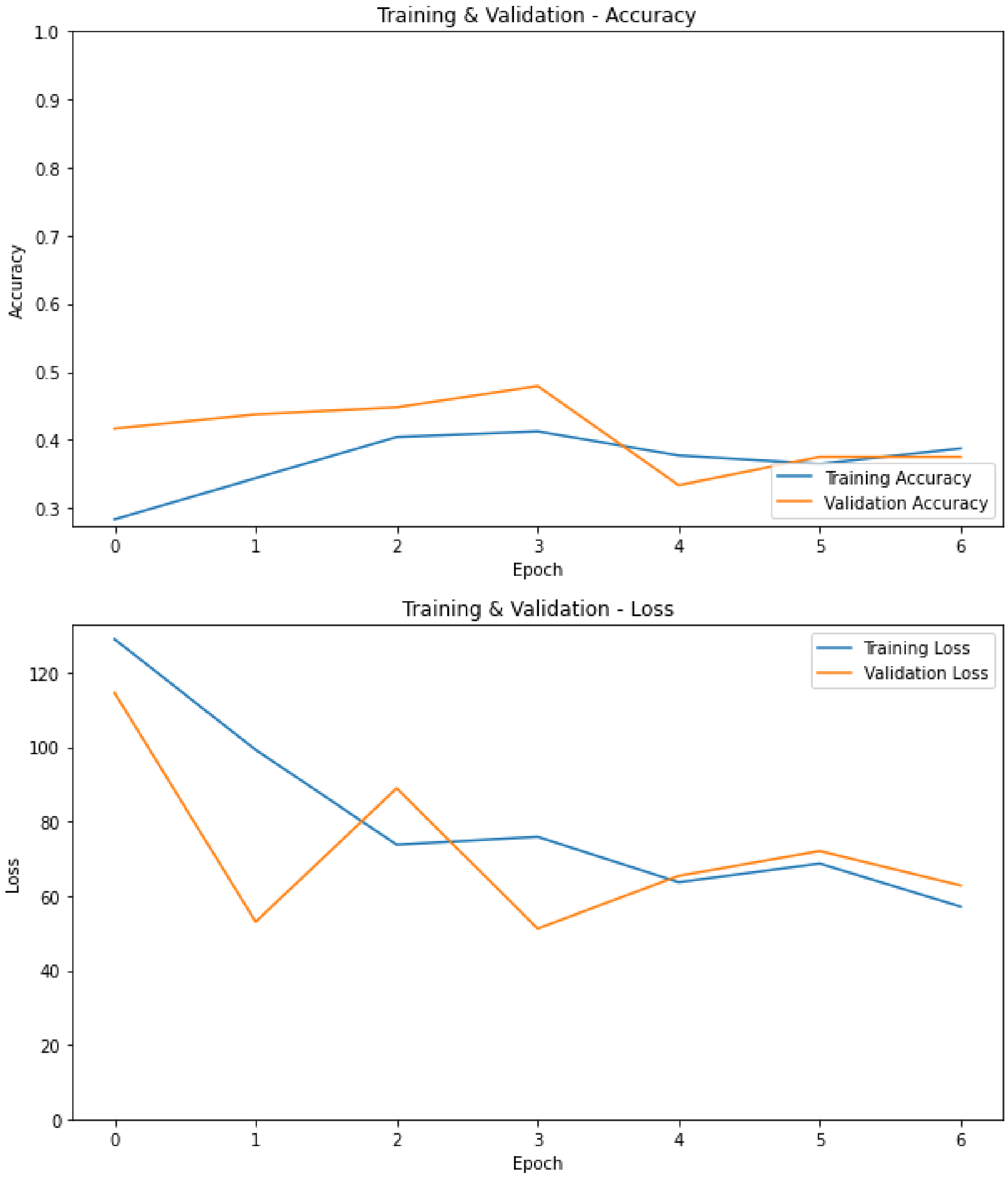


1.8 MobileNETV2

[21]:
```
model_name = PRETRAINED_MODELS[1]
save_model_filepath = 'models/'+ model_name + 'exp3.h5'
logs_dir = 'logs/fit/MobileNetV2exp3'
model = train_and_validate_model(model_name = model_name,
                                 train_generator=sub_sampled_train_gen,
                                 valid_generator=sub_sampled_valid_gen,
                                 save_model_filepath=save_model_filepath,
                                 logs_dir=logs_dir,
                                 num_classes=len(sub_samples))
```

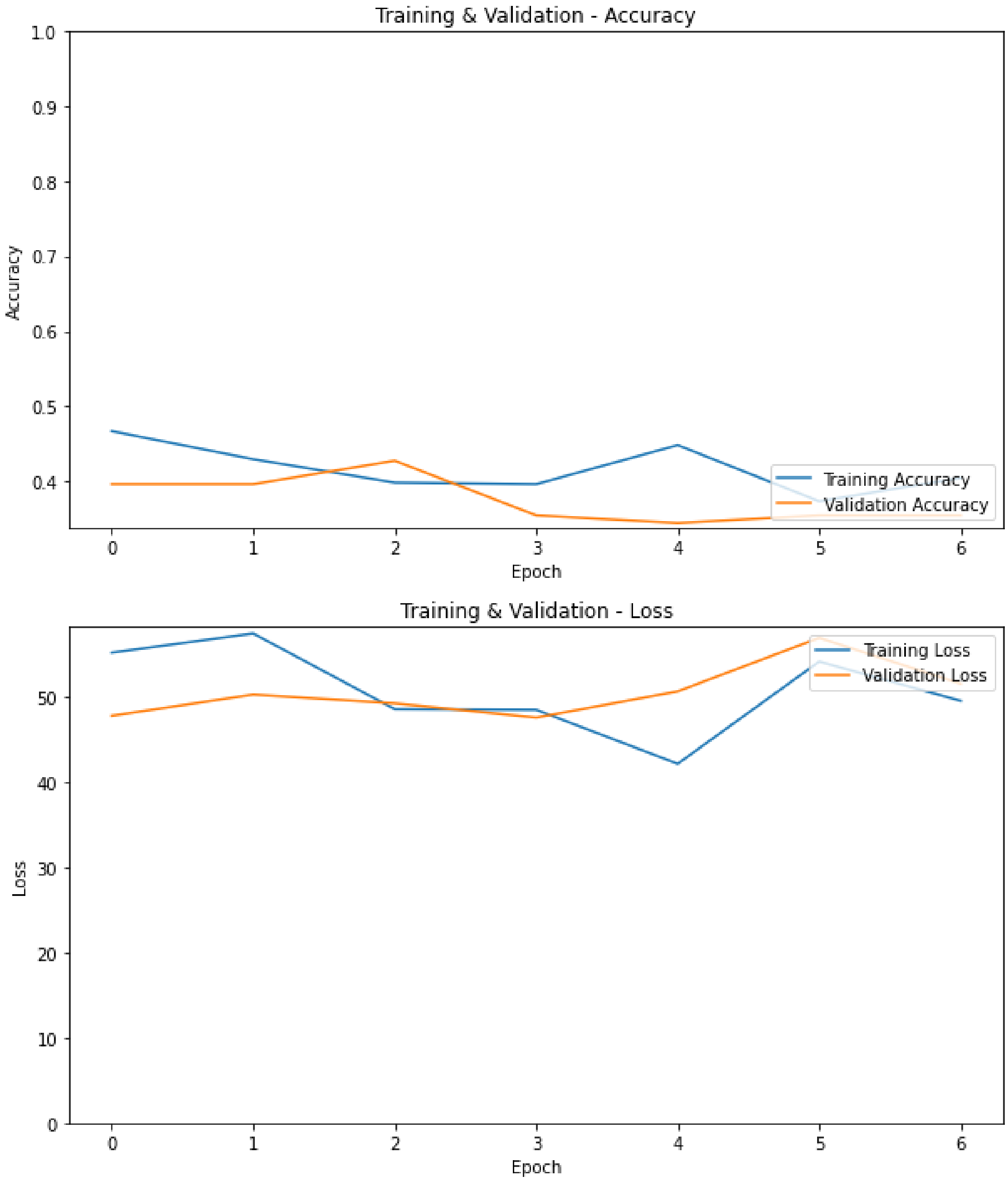
```
MobileNetV2
learning rate 0.01
Downloading MobileNetV2
Epoch 1/10
15/15 [==============================] - 12s 705ms/step - loss: 129.0487 -
accuracy: 0.2833 - val_loss: 114.6313 - val_accuracy: 0.4167

Epoch 00001: val_loss improved from inf to 114.63129, saving model to
models/MobileNetV2exp3.h5
Epoch 2/10
15/15 [==============================] - 9s 615ms/step - loss: 99.3256 -
accuracy: 0.3438 - val_loss: 53.0593 - val_accuracy: 0.4375

Epoch 00002: val_loss improved from 114.63129 to 53.05933, saving model to
models/MobileNetV2exp3.h5
Epoch 3/10
15/15 [==============================] - 9s 613ms/step - loss: 73.8207 -
accuracy: 0.4042 - val_loss: 88.9984 - val_accuracy: 0.4479

Epoch 00003: val_loss did not improve from 53.05933
Epoch 4/10
15/15 [==============================] - 9s 619ms/step - loss: 75.9351 -
accuracy: 0.4125 - val_loss: 51.2674 - val_accuracy: 0.4792

Epoch 00004: val_loss improved from 53.05933 to 51.26743, saving model to
models/MobileNetV2exp3.h5
Epoch 5/10
15/15 [==============================] - 9s 610ms/step - loss: 63.7093 -
accuracy: 0.3771 - val_loss: 65.4302 - val_accuracy: 0.3333

Epoch 00005: val_loss did not improve from 51.26743
Epoch 6/10
15/15 [==============================] - 9s 614ms/step - loss: 68.7459 -
accuracy: 0.3646 - val_loss: 72.1199 - val_accuracy: 0.3750

Epoch 00006: val_loss did not improve from 51.26743
Epoch 7/10
15/15 [==============================] - 9s 611ms/step - loss: 57.1470 -
accuracy: 0.3875 - val_loss: 62.8307 - val_accuracy: 0.3750

Epoch 00007: val_loss did not improve from 51.26743
Restoring model weights from the end of the best epoch.
```

```
Epoch 00007: early stopping
MobileNetV2 Accuracy and Loss plots
```



```
Fine-Tuned MobileNetV2 Training and Validation:
```

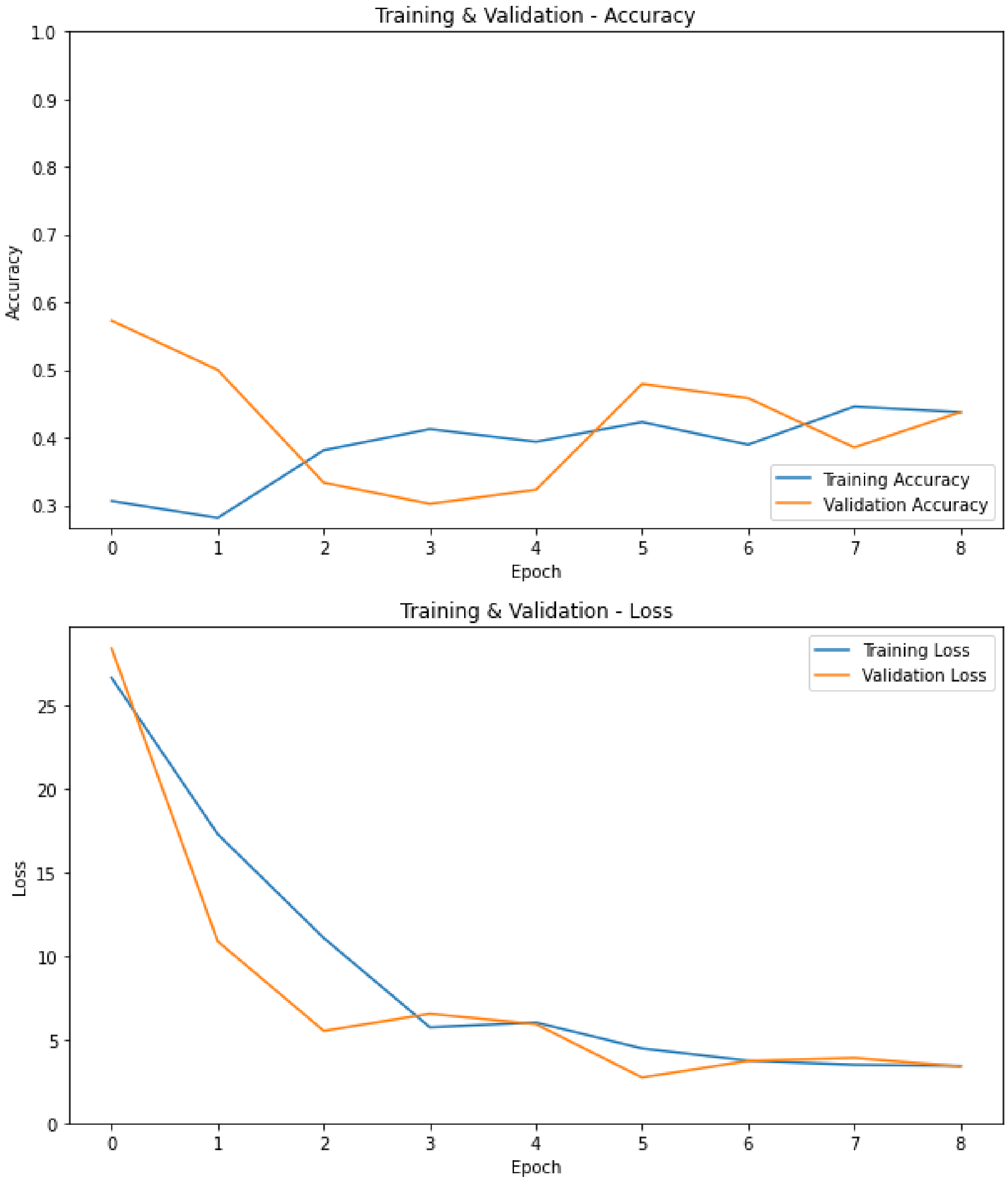
```
Epoch 1/10
15/15 [==============================] - 12s 697ms/step - loss: 55.2296 -
accuracy: 0.4667 - val_loss: 47.8143 - val_accuracy: 0.3958

Epoch 00001: val_loss improved from inf to 47.81433, saving model to
models/my_model.h5
Epoch 2/10
15/15 [==============================] - 9s 609ms/step - loss: 57.4825 -
accuracy: 0.4292 - val_loss: 50.3060 - val_accuracy: 0.3958

Epoch 00002: val_loss did not improve from 47.81433
Epoch 3/10
15/15 [==============================] - 9s 611ms/step - loss: 48.5996 -
accuracy: 0.3979 - val_loss: 49.2789 - val_accuracy: 0.4271

Epoch 00003: val_loss did not improve from 47.81433
Epoch 4/10
15/15 [==============================] - 9s 615ms/step - loss: 48.5192 -
accuracy: 0.3958 - val_loss: 47.6010 - val_accuracy: 0.3542

Epoch 00004: val_loss improved from 47.81433 to 47.60104, saving model to
models/my_model.h5
Epoch 5/10
15/15 [==============================] - 9s 620ms/step - loss: 42.1800 -
accuracy: 0.4479 - val_loss: 50.6820 - val_accuracy: 0.3438

Epoch 00005: val_loss did not improve from 47.60104
Epoch 6/10
15/15 [==============================] - 9s 612ms/step - loss: 54.1863 -
accuracy: 0.3729 - val_loss: 56.9691 - val_accuracy: 0.3542

Epoch 00006: val_loss did not improve from 47.60104
Epoch 7/10
15/15 [==============================] - 9s 616ms/step - loss: 49.5674 -
accuracy: 0.4042 - val_loss: 51.5805 - val_accuracy: 0.3542

Epoch 00007: val_loss did not improve from 47.60104
Restoring model weights from the end of the best epoch.
Epoch 00007: early stopping
Fine-Tuned MobileNetV2 Accuracy and Loss plots
```



1.9 VGG16

[22]:
```
model_name = PRETRAINED_MODELS[2]
save_model_filepath = 'models/'+ model_name + 'exp3.h5'
logs_dir = 'logs/fit/VGG16exp3'
model = train_and_validate_model(model_name = model_name,
                                 train_generator=sub_sampled_train_gen,
                                 valid_generator=sub_sampled_valid_gen,
                                 save_model_filepath=save_model_filepath,
                                 logs_dir=logs_dir,
                                 num_classes=len(sub_samples))
```

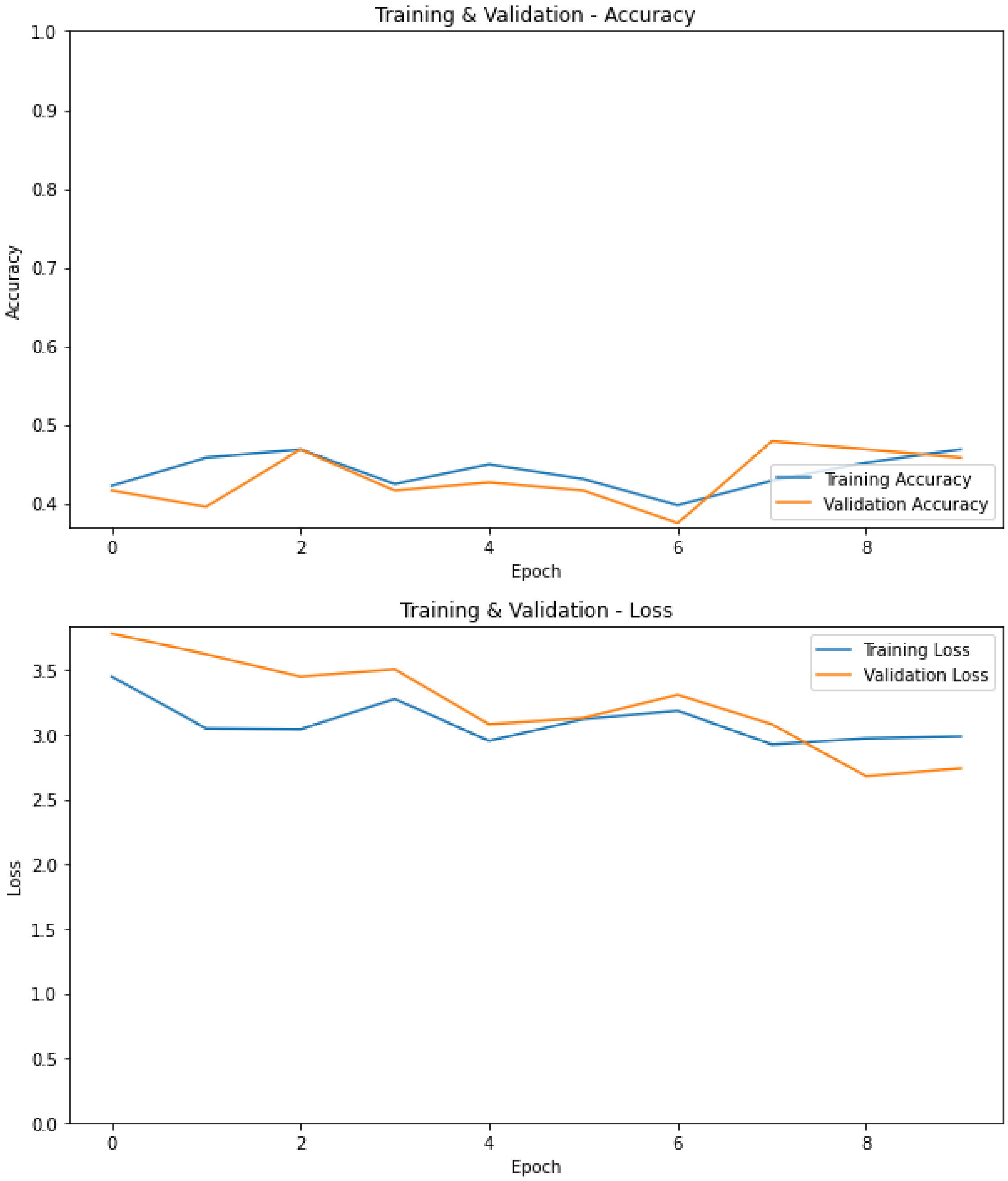
```
VGG16
learning rate 0.01
Downloading VGG16
Epoch 1/10
15/15 [==============================] - 10s 677ms/step - loss: 26.6382 -
accuracy: 0.3063 - val_loss: 28.4014 - val_accuracy: 0.5729

Epoch 00001: val_loss improved from inf to 28.40143, saving model to
models/VGG16exp3.h5
Epoch 2/10
15/15 [==============================] - 9s 618ms/step - loss: 17.2848 -
accuracy: 0.2812 - val_loss: 10.8725 - val_accuracy: 0.5000

Epoch 00002: val_loss improved from 28.40143 to 10.87253, saving model to
models/VGG16exp3.h5
Epoch 3/10
15/15 [==============================] - 9s 623ms/step - loss: 11.0686 -
accuracy: 0.3812 - val_loss: 5.5125 - val_accuracy: 0.3333

Epoch 00003: val_loss improved from 10.87253 to 5.51252, saving model to
models/VGG16exp3.h5
Epoch 4/10
15/15 [==============================] - 9s 619ms/step - loss: 5.7259 -
accuracy: 0.4125 - val_loss: 6.5356 - val_accuracy: 0.3021

Epoch 00004: val_loss did not improve from 5.51252
Epoch 5/10
15/15 [==============================] - 9s 614ms/step - loss: 6.0153 -
accuracy: 0.3938 - val_loss: 5.9145 - val_accuracy: 0.3229

Epoch 00005: val_loss did not improve from 5.51252
Epoch 6/10
15/15 [==============================] - 9s 620ms/step - loss: 4.4626 -
accuracy: 0.4229 - val_loss: 2.7229 - val_accuracy: 0.4792

Epoch 00006: val_loss improved from 5.51252 to 2.72294, saving model to
models/VGG16exp3.h5
Epoch 7/10
15/15 [==============================] - 9s 621ms/step - loss: 3.7286 -
accuracy: 0.3896 - val_loss: 3.7150 - val_accuracy: 0.4583

Epoch 00007: val_loss did not improve from 2.72294
Epoch 8/10
15/15 [==============================] - 9s 619ms/step - loss: 3.4789 -
accuracy: 0.4458 - val_loss: 3.8924 - val_accuracy: 0.3854

Epoch 00008: val_loss did not improve from 2.72294
Epoch 9/10
15/15 [==============================] - 9s 621ms/step - loss: 3.4183 -
accuracy: 0.4375 - val_loss: 3.3850 - val_accuracy: 0.4375

Epoch 00009: val_loss did not improve from 2.72294
Restoring model weights from the end of the best epoch.
Epoch 00009: early stopping
VGG16 Accuracy and Loss plots
```



```
Fine-Tuned VGG16 Training and Validation:
Epoch 1/10
15/15 [==============================] - 11s 753ms/step - loss: 3.4491 -
accuracy: 0.4227 - val_loss: 3.7809 - val_accuracy: 0.4167
```

```
Epoch 00001: val_loss improved from inf to 3.78093, saving model to
models/my_model.h5
Epoch 2/10
15/15 [==============================] - 9s 615ms/step - loss: 3.0471 -
accuracy: 0.4583 - val_loss: 3.6224 - val_accuracy: 0.3958

Epoch 00002: val_loss improved from 3.78093 to 3.62235, saving model to
models/my_model.h5
Epoch 3/10
15/15 [==============================] - 9s 628ms/step - loss: 3.0405 -
accuracy: 0.4688 - val_loss: 3.4493 - val_accuracy: 0.4688

Epoch 00003: val_loss improved from 3.62235 to 3.44932, saving model to
models/my_model.h5
Epoch 4/10
15/15 [==============================] - 9s 615ms/step - loss: 3.2747 -
accuracy: 0.4250 - val_loss: 3.5059 - val_accuracy: 0.4167

Epoch 00004: val_loss did not improve from 3.44932
Epoch 5/10
15/15 [==============================] - 9s 623ms/step - loss: 2.9510 -
accuracy: 0.4500 - val_loss: 3.0788 - val_accuracy: 0.4271

Epoch 00005: val_loss improved from 3.44932 to 3.07880, saving model to
models/my_model.h5
Epoch 6/10
15/15 [==============================] - 9s 620ms/step - loss: 3.1182 -
accuracy: 0.4313 - val_loss: 3.1278 - val_accuracy: 0.4167

Epoch 00006: val_loss did not improve from 3.07880
Epoch 7/10
15/15 [==============================] - 9s 612ms/step - loss: 3.1836 -
accuracy: 0.3979 - val_loss: 3.3077 - val_accuracy: 0.3750

Epoch 00007: val_loss did not improve from 3.07880
Epoch 8/10
15/15 [==============================] - 9s 617ms/step - loss: 2.9240 -
accuracy: 0.4292 - val_loss: 3.0785 - val_accuracy: 0.4792

Epoch 00008: val_loss improved from 3.07880 to 3.07854, saving model to
models/my_model.h5
Epoch 9/10
15/15 [==============================] - 9s 626ms/step - loss: 2.9698 -
accuracy: 0.4521 - val_loss: 2.6795 - val_accuracy: 0.4688

Epoch 00009: val_loss improved from 3.07854 to 2.67950, saving model to
models/my_model.h5
```

```
Epoch 10/10
15/15 [==============================] - 9s 621ms/step - loss: 2.9859 -
accuracy: 0.4688 - val_loss: 2.7403 - val_accuracy: 0.4583

Epoch 00010: val_loss did not improve from 2.67950
Fine-Tuned VGG16 Accuracy and Loss plots
```



1.9.1 Experiment 4: Randoming reducing multiple lables to a single label for an image where multiple lables exist

```
[23]: import random
      reduced_nih_xrays_train_df = nih_xrays_train_df
      reduced_nih_xrays_valid_df = nih_xrays_valid_df
      reduced_nih_xrays_train_df['finding_label'] =␣
       ↪reduced_nih_xrays_train_df['finding_label'].map( lambda x : random.choice(x.
       ↪split('|')) )
      reduced_nih_xrays_valid_df['finding_label'] =␣
       ↪reduced_nih_xrays_valid_df['finding_label'].map( lambda x : random.choice(x.
       ↪split('|')) )
      reduced_sampled_train_gen = train_model.
       ↪get_image_data_generator(reduced_nih_xrays_train_df,batch_size=BATCH_SIZE,image_size=IMAGE_
      reduced_sampled_valid_gen = train_model.
       ↪get_image_data_generator(reduced_nih_xrays_valid_df,batch_size=BATCH_SIZE,image_size=IMAGE_
```

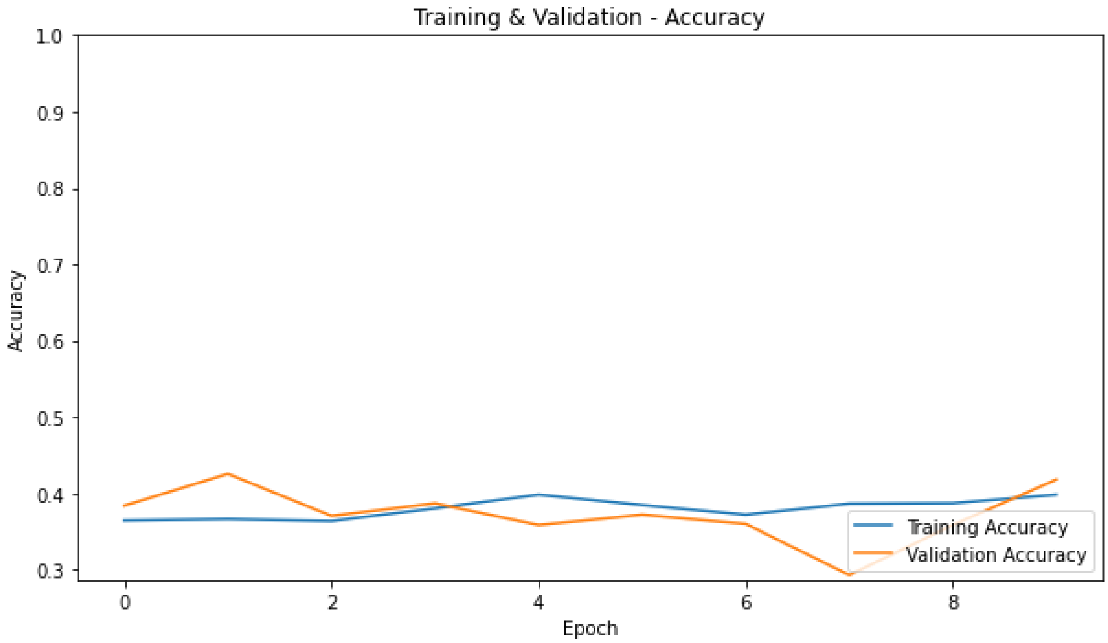
```
Found 89859 validated image filenames belonging to 15 classes.
Found 22245 validated image filenames belonging to 15 classes.
```

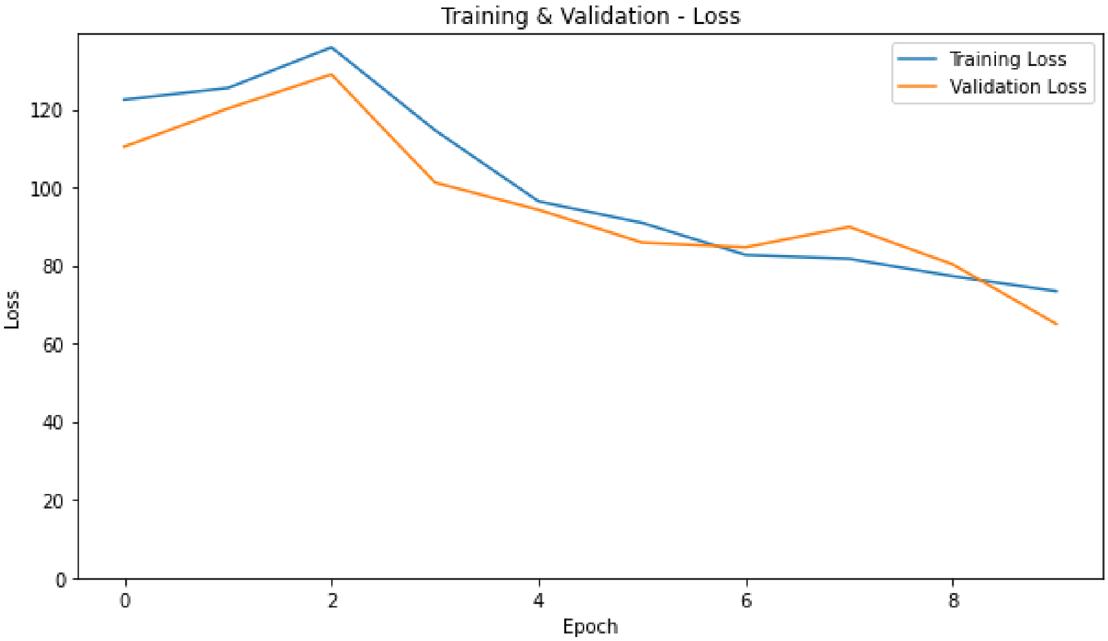
1.9.2 ResNet50V2

```
[24]: model_name = PRETRAINED_MODELS[0]
      save_model_filepath = 'models/'+ model_name + 'exp4.h5'
      logs_dir = 'logs/fit/ResNet50V2exp4'
      model = train_and_validate_model(model_name = model_name,
                                       train_generator=reduced_sampled_train_gen,
                                       valid_generator=reduced_sampled_valid_gen,
                                       save_model_filepath=save_model_filepath,
                                       logs_dir=logs_dir)
```

```
ResNet50V2
learning rate 0.01
Downloading ResNet50V2
Epoch 1/10
87/87 [==============================] - 55s 612ms/step - loss: 122.4626 -
accuracy: 0.3646 - val_loss: 110.4657 - val_accuracy: 0.3839

Epoch 00001: val_loss improved from inf to 110.46574, saving model to
models/ResNet50V2exp4.h5
Epoch 2/10
87/87 [==============================] - 52s 598ms/step - loss: 125.4499 -
accuracy: 0.3664 - val_loss: 120.1959 - val_accuracy: 0.4256

Epoch 00002: val_loss did not improve from 110.46574
Epoch 3/10
87/87 [==============================] - 52s 597ms/step - loss: 135.8712 -
accuracy: 0.3639 - val_loss: 128.9384 - val_accuracy: 0.3705
```

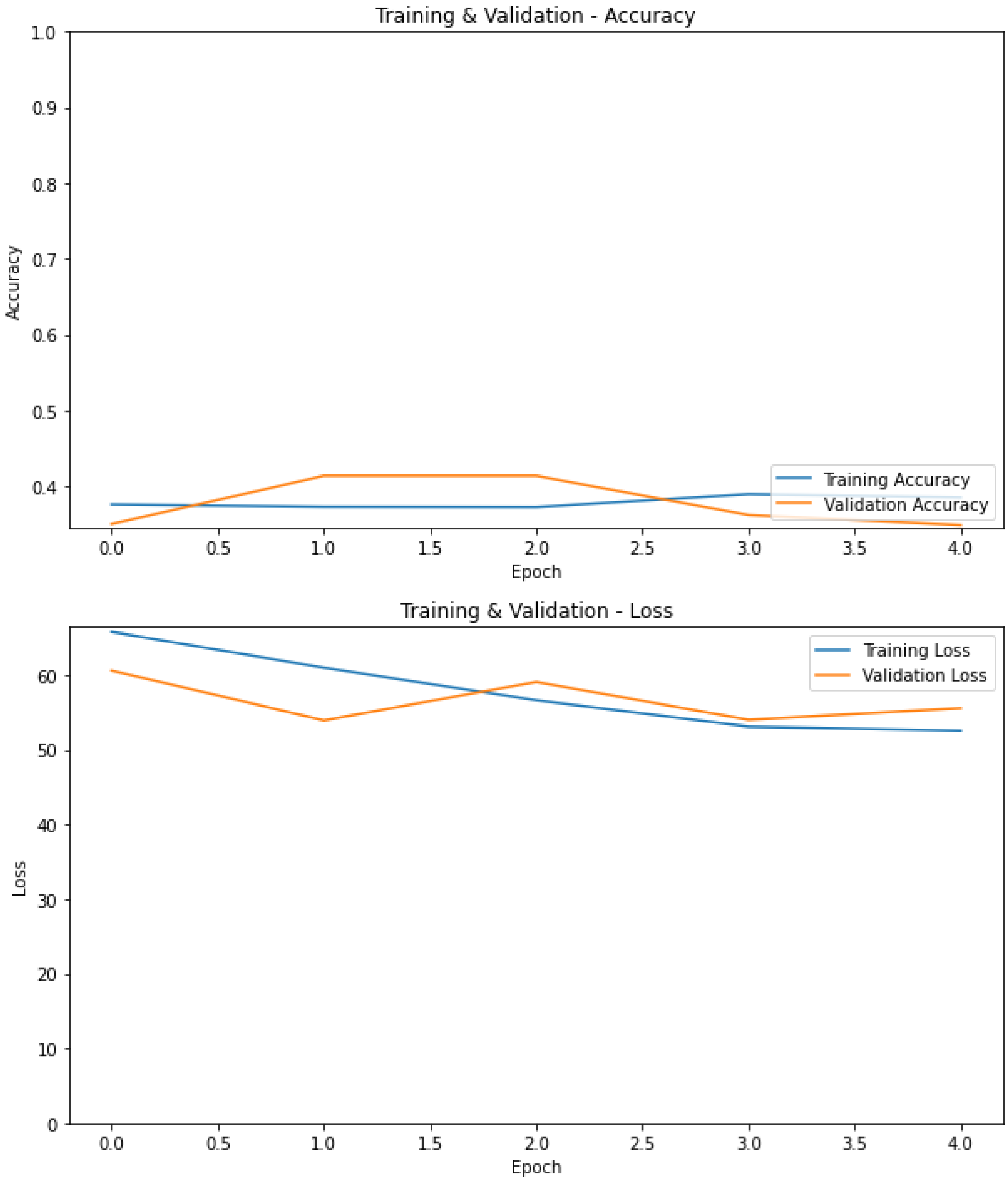
```
Epoch 00003: val_loss did not improve from 110.46574
Epoch 4/10
87/87 [==============================] - 52s 601ms/step - loss: 114.6527 -
accuracy: 0.3804 - val_loss: 101.2422 - val_accuracy: 0.3869

Epoch 00004: val_loss improved from 110.46574 to 101.24219, saving model to
models/ResNet50V2exp4.h5
Epoch 5/10
87/87 [==============================] - 52s 598ms/step - loss: 96.4097 -
accuracy: 0.3983 - val_loss: 94.2430 - val_accuracy: 0.3586

Epoch 00005: val_loss improved from 101.24219 to 94.24298, saving model to
models/ResNet50V2exp4.h5
Epoch 6/10
87/87 [==============================] - 52s 600ms/step - loss: 90.8935 -
accuracy: 0.3847 - val_loss: 85.8948 - val_accuracy: 0.3720

Epoch 00006: val_loss improved from 94.24298 to 85.89484, saving model to
models/ResNet50V2exp4.h5
Epoch 7/10
87/87 [==============================] - 52s 599ms/step - loss: 82.7280 -
accuracy: 0.3721 - val_loss: 84.6953 - val_accuracy: 0.3601

Epoch 00007: val_loss improved from 85.89484 to 84.69525, saving model to
models/ResNet50V2exp4.h5
Epoch 8/10
87/87 [==============================] - 52s 598ms/step - loss: 81.7095 -
accuracy: 0.3865 - val_loss: 89.8863 - val_accuracy: 0.2932

Epoch 00008: val_loss did not improve from 84.69525
Epoch 9/10
87/87 [==============================] - 52s 597ms/step - loss: 77.2685 -
accuracy: 0.3872 - val_loss: 80.2844 - val_accuracy: 0.3586

Epoch 00009: val_loss improved from 84.69525 to 80.28437, saving model to
models/ResNet50V2exp4.h5
Epoch 10/10
87/87 [==============================] - 52s 600ms/step - loss: 73.4267 -
accuracy: 0.3983 - val_loss: 65.0181 - val_accuracy: 0.4182

Epoch 00010: val_loss improved from 80.28437 to 65.01810, saving model to
models/ResNet50V2exp4.h5
ResNet50V2 Accuracy and Loss plots
```





```
Fine-Tuned ResNet50V2 Training and Validation:
Epoch 1/10
87/87 [==============================] - 54s 610ms/step - loss: 65.8133 -
accuracy: 0.3757 - val_loss: 60.6337 - val_accuracy: 0.3497
```

```
Epoch 00001: val_loss improved from inf to 60.63366, saving model to
models/my_model.h5
Epoch 2/10
87/87 [==============================] - 52s 601ms/step - loss: 61.0302 -
accuracy: 0.3725 - val_loss: 53.9360 - val_accuracy: 0.4137

Epoch 00002: val_loss improved from 60.63366 to 53.93600, saving model to
models/my_model.h5
Epoch 3/10
87/87 [==============================] - 52s 598ms/step - loss: 56.6355 -
accuracy: 0.3721 - val_loss: 59.1026 - val_accuracy: 0.4137

Epoch 00003: val_loss did not improve from 53.93600
Epoch 4/10
87/87 [==============================] - 52s 598ms/step - loss: 53.1152 -
accuracy: 0.3894 - val_loss: 54.0174 - val_accuracy: 0.3616

Epoch 00004: val_loss did not improve from 53.93600
Epoch 5/10
87/87 [==============================] - 52s 598ms/step - loss: 52.5876 -
accuracy: 0.3851 - val_loss: 55.5810 - val_accuracy: 0.3482

Epoch 00005: val_loss did not improve from 53.93600
Restoring model weights from the end of the best epoch.
Epoch 00005: early stopping
Fine-Tuned ResNet50V2 Accuracy and Loss plots
```

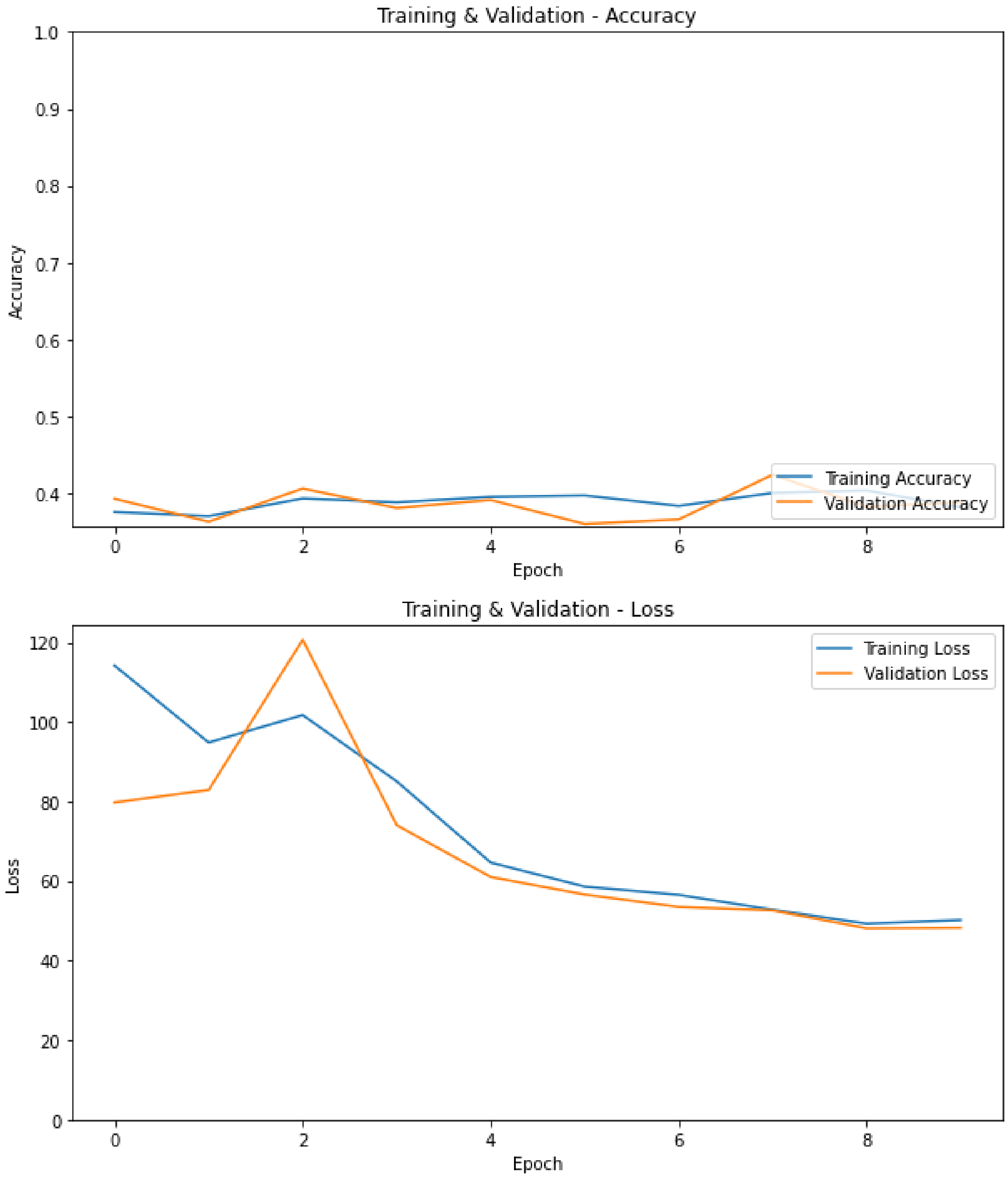


1.10 MobileNETV2

[25]:
```
model_name = PRETRAINED_MODELS[1]
save_model_filepath = 'models/'+ model_name + 'exp4.h5'
logs_dir = 'logs/fit/MobileNetV2exp4'
model = train_and_validate_model(model_name = model_name,
                                 train_generator=reduced_sampled_train_gen,
                                 valid_generator=reduced_sampled_valid_gen,
                                 save_model_filepath=save_model_filepath,
                                 logs_dir=logs_dir)
```

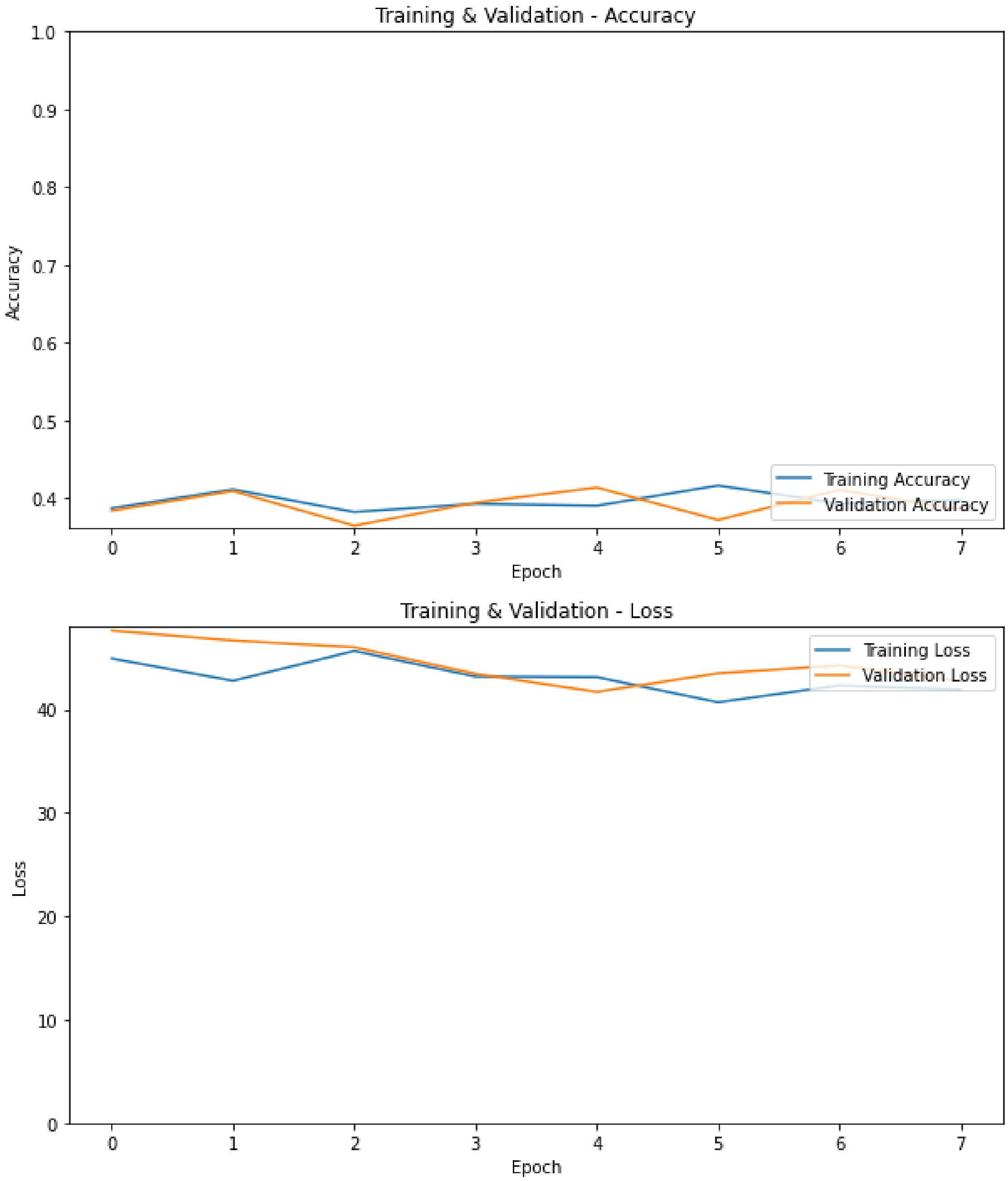
```
MobileNetV2
learning rate 0.01
Downloading MobileNetV2
Epoch 1/10
87/87 [==============================] - 55s 613ms/step - loss: 114.1845 -
accuracy: 0.3757 - val_loss: 79.7889 - val_accuracy: 0.3929

Epoch 00001: val_loss improved from inf to 79.78890, saving model to
models/MobileNetV2exp4.h5
Epoch 2/10
87/87 [==============================] - 52s 597ms/step - loss: 94.8841 -
accuracy: 0.3703 - val_loss: 82.9168 - val_accuracy: 0.3631

Epoch 00002: val_loss did not improve from 79.78890
Epoch 3/10
87/87 [==============================] - 53s 607ms/step - loss: 101.7824 -
accuracy: 0.3933 - val_loss: 120.7274 - val_accuracy: 0.4062

Epoch 00003: val_loss did not improve from 79.78890
Epoch 4/10
87/87 [==============================] - 53s 606ms/step - loss: 85.0962 -
accuracy: 0.3883 - val_loss: 74.0852 - val_accuracy: 0.3810

Epoch 00004: val_loss improved from 79.78890 to 74.08519, saving model to
models/MobileNetV2exp4.h5
Epoch 5/10
87/87 [==============================] - 52s 603ms/step - loss: 64.6731 -
accuracy: 0.3955 - val_loss: 61.0421 - val_accuracy: 0.3914

Epoch 00005: val_loss improved from 74.08519 to 61.04207, saving model to
models/MobileNetV2exp4.h5
Epoch 6/10
87/87 [==============================] - 53s 605ms/step - loss: 58.6234 -
accuracy: 0.3973 - val_loss: 56.6136 - val_accuracy: 0.3601

Epoch 00006: val_loss improved from 61.04207 to 56.61363, saving model to
models/MobileNetV2exp4.h5
Epoch 7/10
87/87 [==============================] - 52s 604ms/step - loss: 56.5407 -
accuracy: 0.3836 - val_loss: 53.5068 - val_accuracy: 0.3661

Epoch 00007: val_loss improved from 56.61363 to 53.50683, saving model to
models/MobileNetV2exp4.h5
```

```
Epoch 8/10
87/87 [==============================] - 53s 605ms/step - loss: 52.7878 -
accuracy: 0.4005 - val_loss: 52.6281 - val_accuracy: 0.4241

Epoch 00008: val_loss improved from 53.50683 to 52.62812, saving model to
models/MobileNetV2exp4.h5
Epoch 9/10
87/87 [==============================] - 52s 602ms/step - loss: 49.3231 -
accuracy: 0.4037 - val_loss: 48.1475 - val_accuracy: 0.3824

Epoch 00009: val_loss improved from 52.62812 to 48.14755, saving model to
models/MobileNetV2exp4.h5
Epoch 10/10
87/87 [==============================] - 53s 605ms/step - loss: 50.2205 -
accuracy: 0.3818 - val_loss: 48.2878 - val_accuracy: 0.3884

Epoch 00010: val_loss did not improve from 48.14755
MobileNetV2 Accuracy and Loss plots
```



```
Fine-Tuned MobileNetV2 Training and Validation:
Epoch 1/10
87/87 [==============================] - 55s 615ms/step - loss: 44.9193 -
accuracy: 0.3872 - val_loss: 47.6128 - val_accuracy: 0.3839
```

```
Epoch 00001: val_loss improved from inf to 47.61280, saving model to
models/my_model.h5
Epoch 2/10
87/87 [==============================] - 53s 606ms/step - loss: 42.7502 -
accuracy: 0.4113 - val_loss: 46.6507 - val_accuracy: 0.4092

Epoch 00002: val_loss improved from 47.61280 to 46.65068, saving model to
models/my_model.h5
Epoch 3/10
87/87 [==============================] - 53s 605ms/step - loss: 45.6410 -
accuracy: 0.3822 - val_loss: 45.9973 - val_accuracy: 0.3646

Epoch 00003: val_loss improved from 46.65068 to 45.99730, saving model to
models/my_model.h5
Epoch 4/10
87/87 [==============================] - 53s 604ms/step - loss: 43.1592 -
accuracy: 0.3930 - val_loss: 43.4295 - val_accuracy: 0.3943

Epoch 00004: val_loss improved from 45.99730 to 43.42955, saving model to
models/my_model.h5
Epoch 5/10
87/87 [==============================] - 52s 603ms/step - loss: 43.1040 -
accuracy: 0.3904 - val_loss: 41.6791 - val_accuracy: 0.4137

Epoch 00005: val_loss improved from 43.42955 to 41.67912, saving model to
models/my_model.h5
Epoch 6/10
87/87 [==============================] - 53s 605ms/step - loss: 40.6700 -
accuracy: 0.4163 - val_loss: 43.4745 - val_accuracy: 0.3720

Epoch 00006: val_loss did not improve from 41.67912
Epoch 7/10
87/87 [==============================] - 52s 601ms/step - loss: 42.3078 -
accuracy: 0.3933 - val_loss: 44.2509 - val_accuracy: 0.4107

Epoch 00007: val_loss did not improve from 41.67912
Epoch 8/10
87/87 [==============================] - 52s 601ms/step - loss: 41.8992 -
accuracy: 0.3980 - val_loss: 42.7245 - val_accuracy: 0.3854

Epoch 00008: val_loss did not improve from 41.67912
Restoring model weights from the end of the best epoch.
Epoch 00008: early stopping
Fine-Tuned MobileNetV2 Accuracy and Loss plots
```

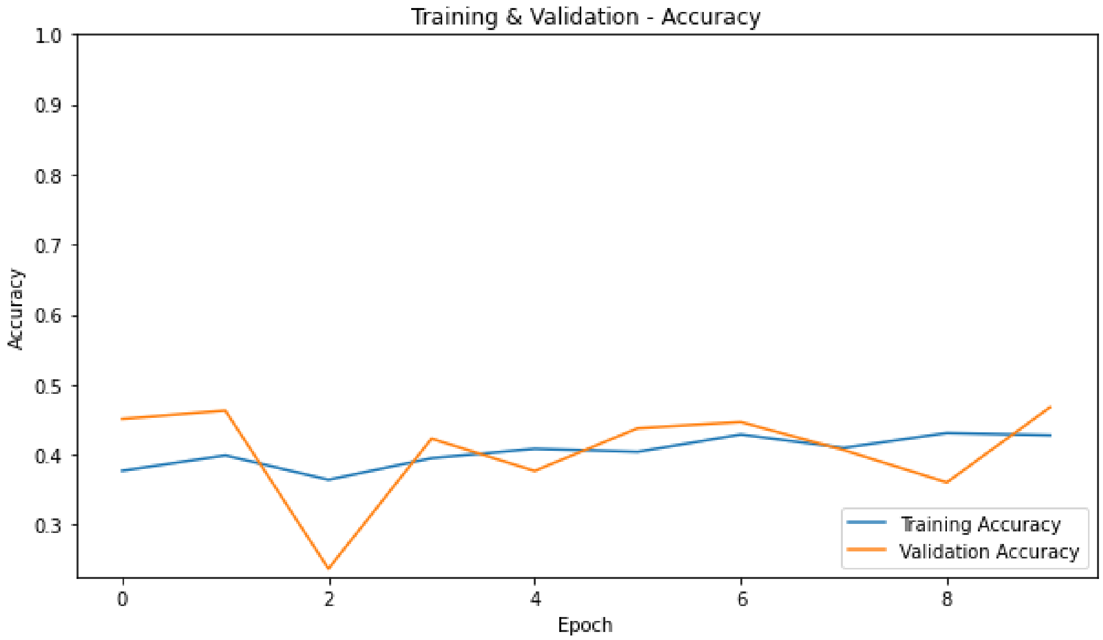


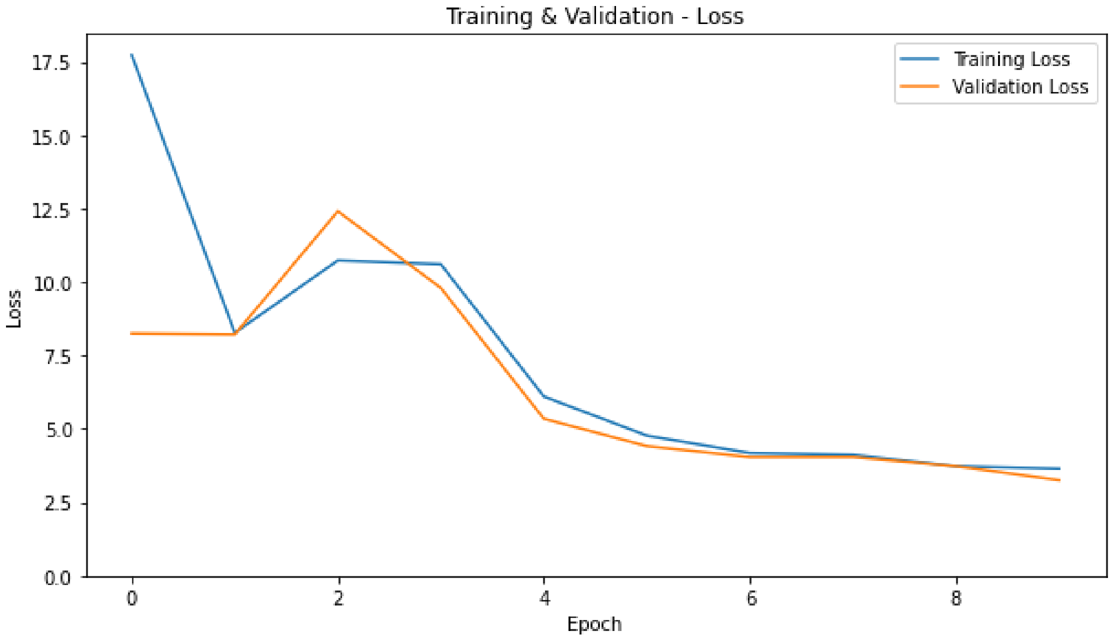
1.11 VGG16

[26]:
```
model_name = PRETRAINED_MODELS[2]
save_model_filepath = 'models/'+ model_name + 'exp4.h5'
logs_dir = 'logs/fit/VGG16exp4'
model = train_and_validate_model(model_name = model_name,
                                 train_generator=reduced_sampled_train_gen,
                                 valid_generator=reduced_sampled_valid_gen,
                                 save_model_filepath=save_model_filepath,
                                 logs_dir=logs_dir)
```

```
VGG16
learning rate 0.01
Downloading VGG16
Epoch 1/10
87/87 [==============================] - 53s 606ms/step - loss: 17.7359 -
accuracy: 0.3768 - val_loss: 8.2508 - val_accuracy: 0.4509

Epoch 00001: val_loss improved from inf to 8.25079, saving model to
models/VGG16exp4.h5
Epoch 2/10
87/87 [==============================] - 52s 596ms/step - loss: 8.2732 -
accuracy: 0.3987 - val_loss: 8.2172 - val_accuracy: 0.4628

Epoch 00002: val_loss improved from 8.25079 to 8.21717, saving model to
models/VGG16exp4.h5
Epoch 3/10
87/87 [==============================] - 52s 594ms/step - loss: 10.7381 -
accuracy: 0.3639 - val_loss: 12.4150 - val_accuracy: 0.2366

Epoch 00003: val_loss did not improve from 8.21717
Epoch 4/10
87/87 [==============================] - 52s 595ms/step - loss: 10.6089 -
accuracy: 0.3948 - val_loss: 9.8033 - val_accuracy: 0.4226

Epoch 00004: val_loss did not improve from 8.21717
Epoch 5/10
87/87 [==============================] - 51s 592ms/step - loss: 6.1056 -
accuracy: 0.4080 - val_loss: 5.3468 - val_accuracy: 0.3765

Epoch 00005: val_loss improved from 8.21717 to 5.34682, saving model to
models/VGG16exp4.h5
Epoch 6/10
87/87 [==============================] - 52s 597ms/step - loss: 4.7719 -
accuracy: 0.4037 - val_loss: 4.4156 - val_accuracy: 0.4375

Epoch 00006: val_loss improved from 5.34682 to 4.41557, saving model to
models/VGG16exp4.h5
Epoch 7/10
87/87 [==============================] - 52s 598ms/step - loss: 4.1801 -
accuracy: 0.4285 - val_loss: 4.0432 - val_accuracy: 0.4464

Epoch 00007: val_loss improved from 4.41557 to 4.04325, saving model to
models/VGG16exp4.h5
```

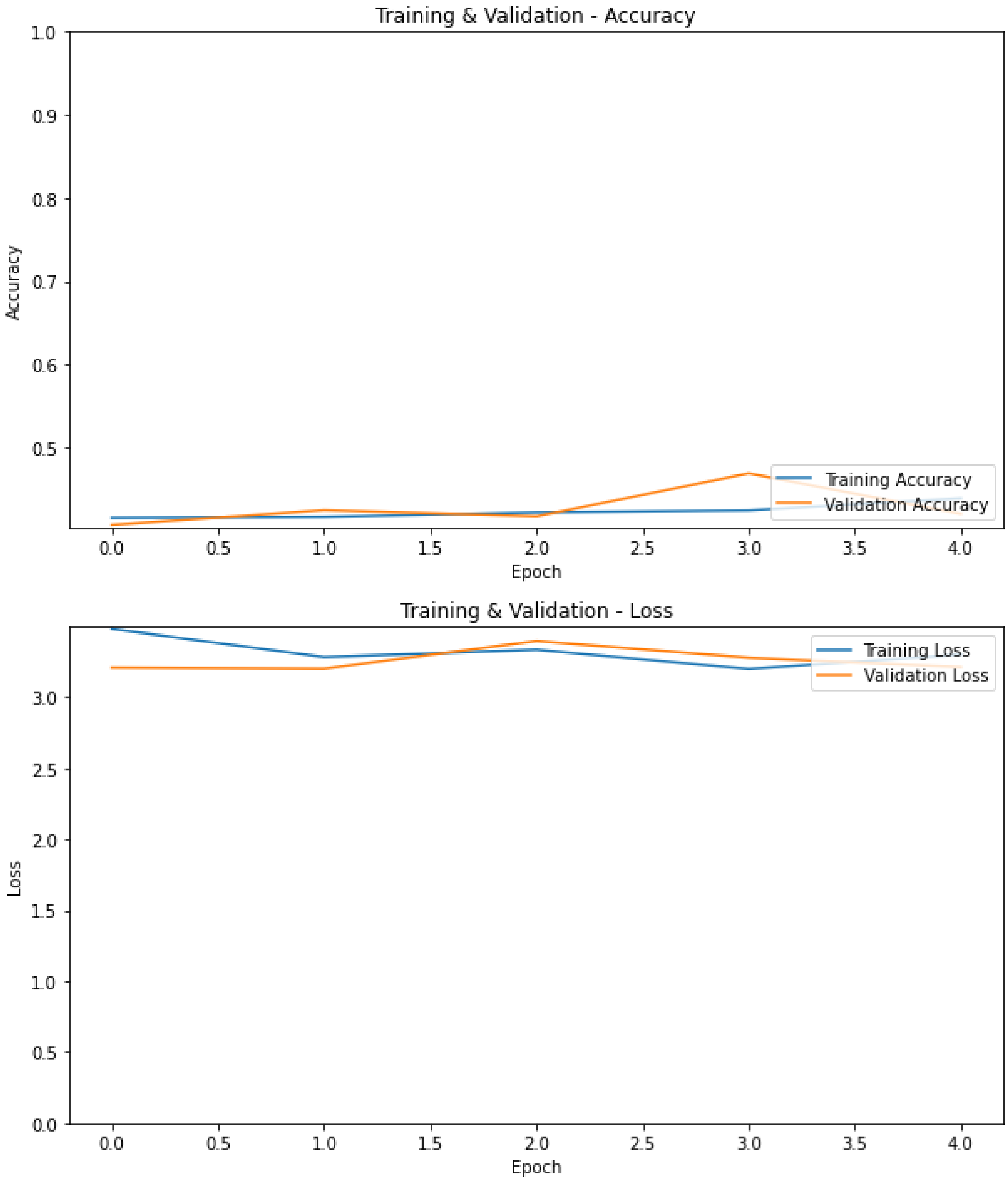
```
Epoch 8/10
87/87 [==============================] - 52s 593ms/step - loss: 4.1201 -
accuracy: 0.4095 - val_loss: 4.0459 - val_accuracy: 0.4062

Epoch 00008: val_loss did not improve from 4.04325
Epoch 9/10
87/87 [==============================] - 52s 597ms/step - loss: 3.7300 -
accuracy: 0.4307 - val_loss: 3.7279 - val_accuracy: 0.3601

Epoch 00009: val_loss improved from 4.04325 to 3.72789, saving model to
models/VGG16exp4.h5
Epoch 10/10
87/87 [==============================] - 52s 601ms/step - loss: 3.6445 -
accuracy: 0.4274 - val_loss: 3.2565 - val_accuracy: 0.4673

Epoch 00010: val_loss improved from 3.72789 to 3.25647, saving model to
models/VGG16exp4.h5
VGG16 Accuracy and Loss plots
```





```
Fine-Tuned VGG16 Training and Validation:
Epoch 1/10
87/87 [==============================] - 53s 607ms/step - loss: 3.4826 -
accuracy: 0.4149 - val_loss: 3.2106 - val_accuracy: 0.4062
```

```
Epoch 00001: val_loss improved from inf to 3.21057, saving model to
models/my_model.h5
Epoch 2/10
87/87 [==============================] - 52s 603ms/step - loss: 3.2851 -
accuracy: 0.4159 - val_loss: 3.2036 - val_accuracy: 0.4241

Epoch 00002: val_loss improved from 3.21057 to 3.20362, saving model to
models/my_model.h5
Epoch 3/10
87/87 [==============================] - 52s 602ms/step - loss: 3.3365 -
accuracy: 0.4213 - val_loss: 3.3981 - val_accuracy: 0.4167

Epoch 00003: val_loss did not improve from 3.20362
Epoch 4/10
87/87 [==============================] - 52s 603ms/step - loss: 3.2017 -
accuracy: 0.4239 - val_loss: 3.2806 - val_accuracy: 0.4688

Epoch 00004: val_loss did not improve from 3.20362
Epoch 5/10
87/87 [==============================] - 52s 604ms/step - loss: 3.3022 -
accuracy: 0.4386 - val_loss: 3.2150 - val_accuracy: 0.4196

Epoch 00005: val_loss did not improve from 3.20362
Restoring model weights from the end of the best epoch.
Epoch 00005: early stopping
Fine-Tuned VGG16 Accuracy and Loss plots
```



```
[27]: %tensorboard --logdir logs/fit/
```

```
<IPython.core.display.HTML object>
```

1.11.1 Clean UP

run this cell after completing execution of the notebook

[28]:

```
# clear gpu memory
from numba import cuda
device = cuda.get_current_device()
device.reset()
```

Run this cell from command prompt

jupyter-nbconvert –to pdf COVID-19-Image-Classification-phase1.ipynb