```
pip install tensorflow
```

```
Requirement already satisfied: tensorflow in
/usr/local/lib/python3.10/dist-packages (2.15.0)
Requirement already satisfied: absl-py>=1.0.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=23.5.26 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (24.3.25)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1
in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.5.4)
Requirement already satisfied: google-pasta>=0.1.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: h5py>=2.9.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (3.9.0)
Requirement already satisfied: libclang>=13.0.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (18.1.1)
Requirement already satisfied: ml-dtypes~=0.2.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: numpy<2.0.0,>=1.23.5 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.25.2)
Requirement already satisfied: opt-einsum>=2.3.2 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (3.3.0)
Requirement already satisfied: packaging in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (24.0)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!
=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (3.20.3)
Requirement already satisfied: setuptools in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (67.7.2)
Requirement already satisfied: six>=1.12.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.4.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (4.11.0)
Requirement already satisfied: wrapt<1.15,>=1.11.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.14.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.36.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.62.2)
Requirement already satisfied: tensorboard<2.16,>=2.15 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.15.2)
Requirement already satisfied: tensorflow-estimator<2.16,>=2.15.0
in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.15.0)
Requirement already satisfied: keras<2.16,>=2.15.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.15.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
```

```
/usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0-
>tensorflow) (0.43.0)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15-
>tensorflow) (2.27.0)
Requirement already satisfied: google-auth-oauthlib<2,>=0.5 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15-
>tensorflow) (1.2.0)
Requirement already satisfied: markdown>=2.6.8 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15-
>tensorflow) (3.6)
Requirement already satisfied: requests<3,>=2.21.0 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15-
>tensorflow) (2.31.0)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0
in /usr/local/lib/python3.10/dist-packages (from
tensorboard<2.16,>=2.15->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15-
>tensorflow) (3.0.2)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard<2.16,>=2.15->tensorflow) (5.3.3)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard<2.16,>=2.15->tensorflow) (0.4.0)
Requirement already satisfied: rsa<5,>=3.1.4 in
/usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard<2.16,>=2.15->tensorflow) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/usr/local/lib/python3.10/dist-packages (from google-auth-
oauthlib<2,>=0.5->tensorboard<2.16,>=2.15->tensorflow) (1.3.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.16,>=2.15->tensorflow) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.16,>=2.15->tensorflow) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.16,>=2.15->tensorflow) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.16,>=2.15->tensorflow) (2024.2.2)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/usr/local/lib/python3.10/dist-packages (from werkzeug>=1.0.1-
>tensorboard<2.16,>=2.15->tensorflow) (2.1.5)
Requirement already satisfied: pyasn1<0.7.0,>=0.4.6 in
/usr/local/lib/python3.10/dist-packages (from pyasn1-modules>=0.2.1-
>google-auth<3,>=1.6.3->tensorboard<2.16,>=2.15->tensorflow) (0.6.0)
Requirement already satisfied: oauthlib>=3.0.0 in
/usr/local/lib/python3.10/dist-packages (from requests-
oauthlib>=0.7.0->google-auth-oauthlib<2,>=0.5-
>tensorboard<2.16,>=2.15->tensorflow) (3.2.2)
```

```
import numpy as np
import os
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications import ResNet50, VGG16, VGG19,
Xception, InceptionV3, DenseNet121
from tensorflow.keras.applications.resnet50 import preprocess_input as
preprocess_input_resnet50
from tensorflow.keras.applications.vgg16 import preprocess_input as
preprocess_input_vgg16
from tensorflow.keras.applications.vgg19 import preprocess_input as
preprocess_input_vgg19
from tensorflow.keras.applications.xception import preprocess_input as
preprocess_input_xception
from tensorflow.keras.applications.inception_v3 import
preprocess_input as preprocess_input_inceptionv3
from tensorflow.keras.applications.densenet import preprocess_input as
preprocess_input_densenet121
import tensorflow as tf
import matplotlib.pyplot as plt
from tqdm.notebook import tqdm
import pandas as pd
import numpy as np
from tensorflow import keras
import cv2
import pickle
from tensorflow.keras import backend as K
from PIL import Image
import matplotlib.patches as mpatches
```

```
from google.colab import files
uploaded = files.upload()
```

```
for filename in uploaded.keys():
  print('User uploaded file "{name}" with length {length}
bytes'.format(
      name=filename, length=len(uploaded[filename])))
```

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

```
Saving rough.rar to rough.rar
User uploaded file "rough.rar" with length 69417355 bytes
```

```
# Install rarfile if not already installed
!pip install rarfile
```

```
Collecting rarfile
  Downloading rarfile-4.2-py3-none-any.whl (29 kB)
Installing collected packages: rarfile
Successfully installed rarfile-4.2
```

```python
import rarfile
import os
import pandas as pd
```

```
!unrar l /content/rough.rar
!mkdir /content/rough.rar
!unrar x -p- /content/rough.rar
```

```
UNRAR 6.11 beta 1 freeware      Copyright (c) 1993-2022 Alexander
Roshal

Archive: /content/rough.rar
Details: RAR 5

 Attributes      Size     Date    Time   Name
----------- ---------  ---------- -----  ----
    ..A....     18831  2023-06-09 23:32  rough/Bird-drop/Bird (1).jpeg
    ..A....     55972  2023-06-09 23:32  rough/Bird-drop/Bird (10).jpg
    ..A....    529769  2023-06-09 23:32  rough/Bird-drop/Bird (11).jpg
    ..A....     17809  2023-06-09 23:32  rough/Bird-drop/Bird (12).jpg
    ..A....     87746  2023-06-09 23:32  rough/Bird-drop/Bird (13).jpg
    ..A....     55229  2023-06-09 23:32  rough/Bird-drop/Bird (14).JPG
    ..A....     66414  2023-06-09 23:32  rough/Bird-drop/Bird (15).JPG
    ..A....     54936  2023-06-09 23:32  rough/Bird-drop/Bird (16).JPG
    ..A....     92765  2023-06-09 23:32  rough/Bird-drop/Bird (17).jpg
    ..A....     36628  2023-06-09 23:32  rough/Bird-drop/Bird (18).jpg
    ..A....     71205  2023-06-09 23:32  rough/Bird-drop/Bird (19).jpg
    ..A....     23869  2023-06-09 23:32  rough/Bird-drop/Bird (2).JPG
    ..A....    102381  2023-06-09 23:32  rough/Bird-drop/Bird (20).jpg
    ..A....   1012730  2023-06-09 23:32  rough/Bird-drop/Bird (3).jpeg
    ..A....     30083  2023-06-09 23:32  rough/Bird-drop/Bird (3).jpg
    ..A....     55972  2023-06-09 23:32  rough/Bird-drop/Bird (4).jpg
    ..A....     63016  2023-06-09 23:32  rough/Bird-drop/Bird (5).jpg
    ..A....     52168  2023-06-09 23:32  rough/Bird-drop/Bird (6).jpg
    ..A....   1969616  2023-06-09 23:32  rough/Bird-drop/Bird (7).jpg
    ..A....     55972  2023-06-09 23:32  rough/Bird-drop/Bird (8).jpg
    ..A....     79907  2023-06-09 23:32  rough/Bird-drop/Bird (9).jpg
    ..A....      5716  2023-06-09 23:32  rough/Bird-drop/desktop.ini
    ..A....     69816  2023-06-09 23:32  rough/Clean/Clean (1).jpeg
    ..A....    167409  2023-06-09 23:32  rough/Clean/Clean (10).JPG
    ..A....    179181  2023-06-09 23:32  rough/Clean/Clean (11).jpg
    ..A....    100172  2023-06-09 23:32  rough/Clean/Clean (12).jpg
    ..A....   2079038  2023-06-09 23:32  rough/Clean/Clean (13).jpg
    ..A....    856600  2023-06-09 23:32  rough/Clean/Clean (14).jpg
    ..A....    178696  2023-06-09 23:32  rough/Clean/Clean (15).JPG
    ..A....    136419  2023-06-09 23:32  rough/Clean/Clean (16).JPG
    ..A....    146028  2023-06-09 23:32  rough/Clean/Clean (17).jpg
    ..A....    150044  2023-06-09 23:32  rough/Clean/Clean (18).jpg
    ..A....   2354955  2023-06-09 23:32  rough/Clean/Clean (19).jpg
    ..A....     82865  2023-06-09 23:32  rough/Clean/Clean (2).jpg
    ..A....   1719877  2023-06-09 23:32  rough/Clean/Clean (20).jpg
    ..A....   1822953  2023-06-09 23:32  rough/Clean/Clean (21).jpg
    ..A....   1125807  2023-06-09 23:32  rough/Clean/Clean (22).jpg
    ..A....   1233515  2023-06-09 23:32  rough/Clean/Clean (23).jpg
    ..A....  12602350  2023-06-09 23:32  rough/Clean/Clean (24).jpg
    ..A....    609528  2023-06-09 23:32  rough/Clean/Clean (25).jpg
    ..A....     89033  2023-06-09 23:32  rough/Clean/Clean (3).jpg
    ..A....    105790  2023-06-09 23:32  rough/Clean/Clean (4).jpg
    ..A....     79123  2023-06-09 23:32  rough/Clean/Clean (5).jpg
    ..A....   1236223  2023-06-09 23:32  rough/Clean/Clean (6).jpg
    ..A....    215152  2023-06-09 23:32  rough/Clean/Clean (7).jpg
    ..A....     99003  2023-06-09 23:32  rough/Clean/Clean (8).jpg
    ..A....     88851  2023-06-09 23:32  rough/Clean/Clean (9).jpg
    ..A....      6606  2023-06-09 23:32  rough/Clean/desktop.ini
    ..A....      6406  2023-06-09 23:32  rough/Dusty/desktop.ini
    ..A....    351812  2023-06-09 23:32  rough/Dusty/Dust (1).jpg
    ..A....     98292  2023-06-09 23:32  rough/Dusty/Dust (10).jpg
    ..A....    979025  2023-06-09 23:32  rough/Dusty/Dust (11).jpg
    ..A....    197413  2023-06-09 23:32  rough/Dusty/Dust (12).jpg
    ..A....     20422  2023-06-09 23:32  rough/Dusty/Dust (13).jpg
    ..A....    320537  2023-06-09 23:32  rough/Dusty/Dust (14).jpg
    ..A....    129097  2023-06-09 23:32  rough/Dusty/Dust (15).jpg
    ..A....     85548  2023-06-09 23:32  rough/Dusty/Dust (16).jpg
    ..A....   5425115  2023-06-09 23:32  rough/Dusty/Dust (17).jpg
    ..A....    165806  2023-06-09 23:32  rough/Dusty/Dust (18).jpg
    ..A....    226938  2023-06-09 23:32  rough/Dusty/Dust (19).jpg
    ..A....    636669  2023-06-09 23:32  rough/Dusty/Dust (2).jpg
    ..A....    198898  2023-06-09 23:32  rough/Dusty/Dust (20).jpg
    ..A....     72979  2023-06-09 23:32  rough/Dusty/Dust (21).jpg
    ..A....     58665  2023-06-09 23:32  rough/Dusty/Dust (22).jpg
    ..A....    979025  2023-06-09 23:32  rough/Dusty/Dust (23).jpg
    ..A....    888273  2023-06-09 23:32  rough/Dusty/Dust (24).jpg
    ..A....   3427228  2023-06-09 23:32  rough/Dusty/Dust (25).jpg
    ..A....   2781221  2023-06-09 23:32  rough/Dusty/Dust (26).jpg
    ..A....    358705  2023-06-09 23:32  rough/Dusty/Dust (27).jpg
    ..A....    146472  2023-06-09 23:32  rough/Dusty/Dust (28).jpg
    ..A....    129744  2023-06-09 23:32  rough/Dusty/Dust (29).jpg
    ..A....    569732  2023-06-09 23:32  rough/Dusty/Dust (3).jpg
    ..A....    383162  2023-06-09 23:32  rough/Dusty/Dust (30).jpg
    ..A....    434968  2023-06-09 23:32  rough/Dusty/Dust (4).jpg
    ..A....     35709  2023-06-09 23:32  rough/Dusty/Dust (5).jpg
    ..A....   3428307  2023-06-09 23:32  rough/Dusty/Dust (6).jpg
    ..A....     61361  2023-06-09 23:32  rough/Dusty/Dust (7).jpg
    ..A....     26169  2023-06-09 23:32  rough/Dusty/Dust (8).jpg
    ..A....    289770  2023-06-09 23:32  rough/Dusty/Dust (9).jpg
    ..A....      3854  2023-06-09 23:32
rough/Electrical-damage/desktop.ini
    ..A....     85252  2023-06-09 23:32
rough/Electrical-damage/Electrical (1).jpg
    ..A....    288961  2023-06-09 23:32
rough/Electrical-damage/Electrical (1).png
    ..A....     47908  2023-06-09 23:32
rough/Electrical-damage/Electrical (10).jpg
    ..A....     46998  2023-06-09 23:32
rough/Electrical-damage/Electrical (11).jpg
    ..A....     19311  2023-06-09 23:32
rough/Electrical-damage/Electrical (12).jpg
    ..A....     27546  2023-06-09 23:32
rough/Electrical-damage/Electrical (13).jpg
    ..A....     68422  2023-06-09 23:32
rough/Electrical-damage/Electrical (14).jpg
    ..A....     34342  2023-06-09 23:32
rough/Electrical-damage/Electrical (15).jpg
    ..A....     30694  2023-06-09 23:32
rough/Electrical-damage/Electrical (16).jpg
    ..A....     36486  2023-06-09 23:32
rough/Electrical-damage/Electrical (17).jpg
    ..A....     31082  2023-06-09 23:32
rough/Electrical-damage/Electrical (18).jpg
    ..A....     31427  2023-06-09 23:32
rough/Electrical-damage/Electrical (19).jpg
    ..A....     32605  2023-06-09 23:32
rough/Electrical-damage/Electrical (2).JPG
    ..A....     95547  2023-06-09 23:32
rough/Electrical-damage/Electrical (2).png
    ..A....    512903  2023-06-09 23:32
rough/Electrical-damage/Electrical (20).jpg
    ..A....     26878  2023-06-09 23:32
rough/Electrical-damage/Electrical (21).jpg
    ..A....     20363  2023-06-09 23:32
rough/Electrical-damage/Electrical (22).jpg
    ..A....     35469  2023-06-09 23:32
rough/Electrical-damage/Electrical (23).jpg
    ..A....     24853  2023-06-09 23:32
rough/Electrical-damage/Electrical (24).jpg
    ..A....     38385  2023-06-09 23:32
rough/Electrical-damage/Electrical (25).jpg
    ..A....     27199  2023-06-09 23:32
rough/Electrical-damage/Electrical (26).jpg
    ..A....     25382  2023-06-09 23:32
rough/Electrical-damage/Electrical (27).jpg
    ..A....    478642  2023-06-09 23:32
```

```
rough/Electrical-damage/Electrical (28).jpg
    ..A....     31082  2023-06-09 23:32
rough/Electrical-damage/Electrical (29).jpg
    ..A....    103402  2023-06-09 23:32
rough/Electrical-damage/Electrical (3).JPG
    ..A....    288961  2023-06-09 23:32
rough/Electrical-damage/Electrical (3).png
    ..A....    111465  2023-06-09 23:32
rough/Electrical-damage/Electrical (30).jpg
    ..A....    226912  2023-06-09 23:32
rough/Electrical-damage/Electrical (31).jpg
    ..A....     24259  2023-06-09 23:32
rough/Electrical-damage/Electrical (32).jpg
    ..A....     14975  2023-06-09 23:32
rough/Electrical-damage/Electrical (33).jpg
    ..A....     28530  2023-06-09 23:32
rough/Electrical-damage/Electrical (4).JPG
    ..A....    312064  2023-06-09 23:32
rough/Electrical-damage/Electrical (4).png
    ..A....     61089  2023-06-09 23:32
rough/Electrical-damage/Electrical (5).jpg
    ..A....     41236  2023-06-09 23:32
rough/Electrical-damage/Electrical (6).jpg
    ..A....     36891  2023-06-09 23:32
rough/Electrical-damage/Electrical (7).jpg
    ..A....     71476  2023-06-09 23:32
rough/Electrical-damage/Electrical (8).jpg
    ..A....     42744  2023-06-09 23:32
rough/Electrical-damage/Electrical (9).jpg
    ..A....      3386  2023-06-09 23:32
rough/Physical-Damage/desktop.ini
    ..A....    104019  2023-06-09 23:32
rough/Physical-Damage/Physical (1).jpg
    ..A....     19633  2023-06-09 23:32
rough/Physical-Damage/Physical (10).jpg
    ..A....     48577  2023-06-09 23:32
rough/Physical-Damage/Physical (11).jpg
    ..A....    179048  2023-06-09 23:32
rough/Physical-Damage/Physical (12).jpg
    ..A....     40289  2023-06-09 23:32
rough/Physical-Damage/Physical (13).jpg
    ..A....     30446  2023-06-09 23:32
rough/Physical-Damage/Physical (14).jpg
    ..A....     41474  2023-06-09 23:32
rough/Physical-Damage/Physical (15).jpg
    ..A....    727740  2023-06-09 23:32
rough/Physical-Damage/Physical (16).jpg
    ..A....    132065  2023-06-09 23:32
rough/Physical-Damage/Physical (17).jpg
```

```
    ..A....    151538  2023-06-09 23:32
rough/Physical-Damage/Physical (18).jpg
    ..A....    122024  2023-06-09 23:32
rough/Physical-Damage/Physical (19).jpg
    ..A....    170035  2023-06-09 23:32
rough/Physical-Damage/Physical (2).jpg
    ..A....    635872  2023-06-09 23:32
rough/Physical-Damage/Physical (20).jpg
    ..A....     78181  2023-06-09 23:32
rough/Physical-Damage/Physical (21).jpg
    ..A....     69813  2023-06-09 23:32
rough/Physical-Damage/Physical (22).jpg
    ..A....    548415  2023-06-09 23:32
rough/Physical-Damage/Physical (23).jpg
    ..A....    452769  2023-06-09 23:32
rough/Physical-Damage/Physical (24).jpg
    ..A....    332984  2023-06-09 23:32
rough/Physical-Damage/Physical (25).jpg
    ..A....    119336  2023-06-09 23:32
rough/Physical-Damage/Physical (3).jpg
    ..A....     37652  2023-06-09 23:32
rough/Physical-Damage/Physical (4).jpg
    ..A....     59640  2023-06-09 23:32
rough/Physical-Damage/Physical (5).jpg
    ..A....    170680  2023-06-09 23:32
rough/Physical-Damage/Physical (6).jpg
    ..A....    126921  2023-06-09 23:32
rough/Physical-Damage/Physical (7).jpg
    ..A....    288958  2023-06-09 23:32
rough/Physical-Damage/Physical (8).jpg
    ..A....     43413  2023-06-09 23:32
rough/Physical-Damage/Physical (9).jpg
    ..A....      3710  2023-06-09 23:32
rough/Snow-Covered/desktop.ini
    ..A....    339669  2023-06-09 23:32  rough/Snow-Covered/Snow
(1).jpg
    ..A....     37664  2023-06-09 23:32  rough/Snow-Covered/Snow
(10).jpg
    ..A....    839443  2023-06-09 23:32  rough/Snow-Covered/Snow
(11).jpg
    ..A....    590653  2023-06-09 23:32  rough/Snow-Covered/Snow
(12).jpg
    ..A....     71381  2023-06-09 23:32  rough/Snow-Covered/Snow
(13).jpg
    ..A....     34022  2023-06-09 23:32  rough/Snow-Covered/Snow
(14).jpg
    ..A....     80630  2023-06-09 23:32  rough/Snow-Covered/Snow
(15).jpg
    ..A....    143919  2023-06-09 23:32  rough/Snow-Covered/Snow
```

```
(16).jpg
    ..A....     80613  2023-06-09 23:32  rough/Snow-Covered/Snow
(17).jpg
    ..A....     74389  2023-06-09 23:32  rough/Snow-Covered/Snow
(18).jpg
    ..A....     92251  2023-06-09 23:32  rough/Snow-Covered/Snow
(19).jpg
    ..A....    213291  2023-06-09 23:32  rough/Snow-Covered/Snow
(2).jpg
    ..A....    199091  2023-06-09 23:32  rough/Snow-Covered/Snow
(20).jpg
    ..A....     41273  2023-06-09 23:32  rough/Snow-Covered/Snow
(21).jpg
    ..A....     68246  2023-06-09 23:32  rough/Snow-Covered/Snow
(22).jpg
    ..A....     75705  2023-06-09 23:32  rough/Snow-Covered/Snow
(23).jpg
    ..A....     15637  2023-06-09 23:32  rough/Snow-Covered/Snow
(24).jpg
    ..A....    114253  2023-06-09 23:32  rough/Snow-Covered/Snow
(25).jpg
    ..A....     58499  2023-06-09 23:32  rough/Snow-Covered/Snow
(26).jpg
    ..A....     24547  2023-06-09 23:32  rough/Snow-Covered/Snow
(27).jpg
    ..A....     94537  2023-06-09 23:32  rough/Snow-Covered/Snow
(28).JPG
    ..A....    103826  2023-06-09 23:32  rough/Snow-Covered/Snow
(29).JPG
    ..A....    959571  2023-06-09 23:32  rough/Snow-Covered/Snow
(3).jpg
    ..A....    112911  2023-06-09 23:32  rough/Snow-Covered/Snow
(30).JPG
    ..A....     68548  2023-06-09 23:32  rough/Snow-Covered/Snow
(31).JPG
    ..A....     52767  2023-06-09 23:32  rough/Snow-Covered/Snow
(32).JPG
    ..A....     38476  2023-06-09 23:32  rough/Snow-Covered/Snow
(33).JPG
    ..A....    188519  2023-06-09 23:32  rough/Snow-Covered/Snow
(34).jpg
    ..A....     74375  2023-06-09 23:32  rough/Snow-Covered/Snow
(35).JPG
    ..A....    120379  2023-06-09 23:32  rough/Snow-Covered/Snow
(36).JPG
    ..A....    132581  2023-06-09 23:32  rough/Snow-Covered/Snow
(37).jpg
    ..A....    640259  2023-06-09 23:32  rough/Snow-Covered/Snow
(4).jpg
```

```
    ..A....     56027  2023-06-09 23:32  rough/Snow-Covered/Snow
(5).jpg
    ..A....    127349  2023-06-09 23:32  rough/Snow-Covered/Snow
(6).jpg
    ..A....     98829  2023-06-09 23:32  rough/Snow-Covered/Snow
(7).jpg
    ..A....    770536  2023-06-09 23:32  rough/Snow-Covered/Snow
(8).jpg
    ..A....    118351  2023-06-09 23:32  rough/Snow-Covered/Snow
(9).jpg
    ...D...         0  2024-04-25 10:22  rough/Bird-drop
    ...D...         0  2024-04-25 10:22  rough/Clean
    ...D...         0  2024-04-25 10:22  rough/Dusty
    ...D...         0  2024-04-25 10:23  rough/Electrical-damage
    ...D...         0  2024-04-25 10:23  rough/Physical-Damage
    ...D...         0  2024-04-25 10:23  rough/Snow-Covered
    ...D...         0  2024-04-25 10:53  rough
----------- ---------  ---------- -----  ----
             70144466                    188

mkdir: cannot create directory ‘/content/rough.rar’: File exists

UNRAR 6.11 beta 1 freeware      Copyright (c) 1993-2022 Alexander
Roshal


Extracting from /content/rough.rar

Creating    rough
OK
Creating    rough/Bird-drop
OK
Extracting  rough/Bird-drop/Bird (1).jpeg
0%  OK
Extracting  rough/Bird-drop/Bird (10).jpg
0%  OK
Extracting  rough/Bird-drop/Bird (11).jpg
0%  OK
Extracting  rough/Bird-drop/Bird (12).jpg
0%  OK
Extracting  rough/Bird-drop/Bird (13).jpg
1%  OK
Extracting  rough/Bird-drop/Bird (14).JPG
1%  OK
Extracting  rough/Bird-drop/Bird (15).JPG
1%  OK
Extracting  rough/Bird-drop/Bird (16).JPG
1%  OK
Extracting  rough/Bird-drop/Bird (17).jpg
1%  OK
```

```
Extracting  rough/Bird-drop/Bird (18).jpg
1%  OK
Extracting  rough/Bird-drop/Bird (19).jpg
1%  OK
Extracting  rough/Bird-drop/Bird (2).JPG
1%  OK
Extracting  rough/Bird-drop/Bird (20).jpg
1%  OK
Extracting  rough/Bird-drop/Bird (3).jpeg
3%  OK
Extracting  rough/Bird-drop/Bird (3).jpg
3%  OK
Extracting  rough/Bird-drop/Bird (4).jpg
3%  OK
Extracting  rough/Bird-drop/Bird (5).jpg
3%  OK
Extracting  rough/Bird-drop/Bird (6).jpg
3%  OK
Extracting  rough/Bird-drop/Bird (7).jpg
6%  OK
Extracting  rough/Bird-drop/Bird (8).jpg
6%  OK
Extracting  rough/Bird-drop/Bird (9).jpg
6%  OK
Extracting  rough/Bird-drop/desktop.ini
6%  OK
Creating    rough/Clean
OK
Extracting  rough/Clean/Clean (1).jpeg
6%  OK
Extracting  rough/Clean/Clean (10).JPG
6%  OK
Extracting  rough/Clean/Clean (11).jpg
7%  OK
Extracting  rough/Clean/Clean (12).jpg
7%  OK
Extracting  rough/Clean/Clean (13).jpg
10%  OK
Extracting  rough/Clean/Clean (14).jpg
11%  OK
Extracting  rough/Clean/Clean (15).JPG
11%  OK
Extracting  rough/Clean/Clean (16).JPG
11%  OK
Extracting  rough/Clean/Clean (17).jpg
11%  OK
Extracting  rough/Clean/Clean (18).jpg
12%  OK
Extracting  rough/Clean/Clean (19).jpg
```

```
15%  OK
Extracting  rough/Clean/Clean (2).jpg
15%  OK
Extracting  rough/Clean/Clean (20).jpg
18%  OK
Extracting  rough/Clean/Clean (21).jpg
20%  OK
Extracting  rough/Clean/Clean (22).jpg
22%  OK
Extracting  rough/Clean/Clean (23).jpg
23% 24%  OK
Extracting  rough/Clean/Clean (24).jpg
30% 36% 42%  OK
Extracting  rough/Clean/Clean (25).jpg
42%  OK
Extracting  rough/Clean/Clean (3).jpg
43%  OK
Extracting  rough/Clean/Clean (4).jpg
43%  OK
Extracting  rough/Clean/Clean (5).jpg
43%  OK
Extracting  rough/Clean/Clean (6).jpg
44% 45%  OK
Extracting  rough/Clean/Clean (7).jpg
45%  OK
Extracting  rough/Clean/Clean (8).jpg
45%  OK
Extracting  rough/Clean/Clean (9).jpg
45%  OK
Extracting  rough/Clean/desktop.ini
45%  OK
Creating    rough/Dusty
OK
Extracting  rough/Dusty/desktop.ini
45%  OK
Extracting  rough/Dusty/Dust (1).jpg
46%  OK
Extracting  rough/Dusty/Dust (10).jpg
46%  OK
Extracting  rough/Dusty/Dust (11).jpg
47%  OK
Extracting  rough/Dusty/Dust (12).jpg
48%  OK
Extracting  rough/Dusty/Dust (13).jpg
48%  OK
Extracting  rough/Dusty/Dust (14).jpg
48%  OK
Extracting  rough/Dusty/Dust (15).jpg
48%  OK
```

```
Extracting  rough/Dusty/Dust (16).jpg
48%  OK
Extracting  rough/Dusty/Dust (17).jpg
54% 56%  OK
Extracting  rough/Dusty/Dust (18).jpg
56%  OK
Extracting  rough/Dusty/Dust (19).jpg
57%  OK
Extracting  rough/Dusty/Dust (2).jpg
58%  OK
Extracting  rough/Dusty/Dust (20).jpg
58%  OK
Extracting  rough/Dusty/Dust (21).jpg
58%  OK
Extracting  rough/Dusty/Dust (22).jpg
58%  OK
Extracting  rough/Dusty/Dust (23).jpg
59%  OK
Extracting  rough/Dusty/Dust (24).jpg
61%  OK
Extracting  rough/Dusty/Dust (25).jpg
66%  OK
Extracting  rough/Dusty/Dust (26).jpg
70%  OK
Extracting  rough/Dusty/Dust (27).jpg
70%  OK
Extracting  rough/Dusty/Dust (28).jpg
70%  OK
Extracting  rough/Dusty/Dust (29).jpg
71%  OK
Extracting  rough/Dusty/Dust (3).jpg
71%  OK
Extracting  rough/Dusty/Dust (30).jpg
72%  OK
Extracting  rough/Dusty/Dust (4).jpg
73%  OK
Extracting  rough/Dusty/Dust (5).jpg
73%  OK
Extracting  rough/Dusty/Dust (6).jpg
74% 76% 77% 78%  OK
Extracting  rough/Dusty/Dust (7).jpg
78%  OK
Extracting  rough/Dusty/Dust (8).jpg
78%  OK
Extracting  rough/Dusty/Dust (9).jpg
78%  OK
Creating    rough/Electrical-damage
OK
Extracting  rough/Electrical-damage/desktop.ini
```

```
78%  OK
Extracting  rough/Electrical-damage/Electrical (1).jpg
78%  OK
Extracting  rough/Electrical-damage/Electrical (1).png
79%  OK
Extracting  rough/Electrical-damage/Electrical (10).jpg
79%  OK
Extracting  rough/Electrical-damage/Electrical (11).jpg
79%  OK
Extracting  rough/Electrical-damage/Electrical (12).jpg
79%  OK
Extracting  rough/Electrical-damage/Electrical (13).jpg
79%  OK
Extracting  rough/Electrical-damage/Electrical (14).jpg
79%  OK
Extracting  rough/Electrical-damage/Electrical (15).jpg
79%  OK
Extracting  rough/Electrical-damage/Electrical (16).jpg
79%  OK
Extracting  rough/Electrical-damage/Electrical (17).jpg
79%  OK
Extracting  rough/Electrical-damage/Electrical (18).jpg
79%  OK
Extracting  rough/Electrical-damage/Electrical (19).jpg
79%  OK
Extracting  rough/Electrical-damage/Electrical (2).JPG
79%  OK
Extracting  rough/Electrical-damage/Electrical (2).png
79%  OK
Extracting  rough/Electrical-damage/Electrical (20).jpg
80%  OK
Extracting  rough/Electrical-damage/Electrical (21).jpg
80%  OK
Extracting  rough/Electrical-damage/Electrical (22).jpg
80%  OK
Extracting  rough/Electrical-damage/Electrical (23).jpg
80%  OK
Extracting  rough/Electrical-damage/Electrical (24).jpg
80%  OK
Extracting  rough/Electrical-damage/Electrical (25).jpg
80%  OK
Extracting  rough/Electrical-damage/Electrical (26).jpg
80%  OK
Extracting  rough/Electrical-damage/Electrical (27).jpg
80%  OK
Extracting  rough/Electrical-damage/Electrical (28).jpg
81%  OK
Extracting  rough/Electrical-damage/Electrical (29).jpg
81%  OK
Extracting  rough/Electrical-damage/Electrical (3).JPG
```

```
81%  OK
Extracting  rough/Electrical-damage/Electrical (3).png
81%  OK
Extracting  rough/Electrical-damage/Electrical (30).jpg
82%  OK
Extracting  rough/Electrical-damage/Electrical (31).jpg
82%  OK
Extracting  rough/Electrical-damage/Electrical (32).jpg
82%  OK
Extracting  rough/Electrical-damage/Electrical (33).jpg
82%  OK
Extracting  rough/Electrical-damage/Electrical (4).JPG
82%  OK
Extracting  rough/Electrical-damage/Electrical (4).png
83%  OK
Extracting  rough/Electrical-damage/Electrical (5).jpg
83%  OK
Extracting  rough/Electrical-damage/Electrical (6).jpg
83%  OK
Extracting  rough/Electrical-damage/Electrical (7).jpg
83%  OK
Extracting  rough/Electrical-damage/Electrical (8).jpg
83%  OK
Extracting  rough/Electrical-damage/Electrical (9).jpg
83%  OK
Creating    rough/Physical-Damage
OK
Extracting  rough/Physical-Damage/desktop.ini
83%  OK
Extracting  rough/Physical-Damage/Physical (1).jpg
83%  OK
Extracting  rough/Physical-Damage/Physical (10).jpg
83%  OK
Extracting  rough/Physical-Damage/Physical (11).jpg
83%  OK
Extracting  rough/Physical-Damage/Physical (12).jpg
83%  OK
Extracting  rough/Physical-Damage/Physical (13).jpg
83%  OK
Extracting  rough/Physical-Damage/Physical (14).jpg
83%  OK
Extracting  rough/Physical-Damage/Physical (15).jpg
84%  OK
Extracting  rough/Physical-Damage/Physical (16).jpg
85%  OK
Extracting  rough/Physical-Damage/Physical (17).jpg
85%  OK
Extracting  rough/Physical-Damage/Physical (18).jpg
85%  OK
```

```
Extracting  rough/Physical-Damage/Physical (19).jpg
85%  OK
Extracting  rough/Physical-Damage/Physical (2).jpg
85%  OK
Extracting  rough/Physical-Damage/Physical (20).jpg
86%  OK
Extracting  rough/Physical-Damage/Physical (21).jpg
86%  OK
Extracting  rough/Physical-Damage/Physical (22).jpg
87%  OK
Extracting  rough/Physical-Damage/Physical (23).jpg
87%  OK
Extracting  rough/Physical-Damage/Physical (24).jpg
88%  OK
Extracting  rough/Physical-Damage/Physical (25).jpg
88%  OK
Extracting  rough/Physical-Damage/Physical (3).jpg
89%  OK
Extracting  rough/Physical-Damage/Physical (4).jpg
89%  OK
Extracting  rough/Physical-Damage/Physical (5).jpg
89%  OK
Extracting  rough/Physical-Damage/Physical (6).jpg
89%  OK
Extracting  rough/Physical-Damage/Physical (7).jpg
89%  OK
Extracting  rough/Physical-Damage/Physical (8).jpg
90%  OK
Extracting  rough/Physical-Damage/Physical (9).jpg
90%  OK
Creating    rough/Snow-Covered
OK
Extracting  rough/Snow-Covered/desktop.ini
90%  OK
Extracting  rough/Snow-Covered/Snow (1).jpg
90%  OK
Extracting  rough/Snow-Covered/Snow (10).jpg
90%  OK
Extracting  rough/Snow-Covered/Snow (11).jpg
91%  OK
Extracting  rough/Snow-Covered/Snow (12).jpg
92%  OK
Extracting  rough/Snow-Covered/Snow (13).jpg
92%  OK
Extracting  rough/Snow-Covered/Snow (14).jpg
92%  OK
Extracting  rough/Snow-Covered/Snow (15).jpg
92%  OK
Extracting  rough/Snow-Covered/Snow (16).jpg
```

```
93%  OK
Extracting  rough/Snow-Covered/Snow (17).jpg
93%  OK
Extracting  rough/Snow-Covered/Snow (18).jpg
93%  OK
Extracting  rough/Snow-Covered/Snow (19).jpg
93%  OK
Extracting  rough/Snow-Covered/Snow (2).jpg
93%  OK
Extracting  rough/Snow-Covered/Snow (20).jpg
94%  OK
Extracting  rough/Snow-Covered/Snow (21).jpg
94%  OK
Extracting  rough/Snow-Covered/Snow (22).jpg
94%  OK
Extracting  rough/Snow-Covered/Snow (23).jpg
94%  OK
Extracting  rough/Snow-Covered/Snow (24).jpg
94%  OK
Extracting  rough/Snow-Covered/Snow (25).jpg
94%  OK
Extracting  rough/Snow-Covered/Snow (26).jpg
94%  OK
Extracting  rough/Snow-Covered/Snow (27).jpg
94%  OK
Extracting  rough/Snow-Covered/Snow (28).JPG
94%  OK
Extracting  rough/Snow-Covered/Snow (29).JPG
94%  OK
Extracting  rough/Snow-Covered/Snow (3).jpg
96%  OK
Extracting  rough/Snow-Covered/Snow (30).JPG
96%  OK
Extracting  rough/Snow-Covered/Snow (31).JPG
96%  OK
Extracting  rough/Snow-Covered/Snow (32).JPG
96%  OK
Extracting  rough/Snow-Covered/Snow (33).JPG
96%  OK
Extracting  rough/Snow-Covered/Snow (34).jpg
96%  OK
Extracting  rough/Snow-Covered/Snow (35).JPG
97%  OK
Extracting  rough/Snow-Covered/Snow (36).JPG
97%  OK
Extracting  rough/Snow-Covered/Snow (37).jpg
97%  OK
Extracting  rough/Snow-Covered/Snow (4).jpg
98%  OK
```

```
Extracting  rough/Snow-Covered/Snow (5).jpg
98%  OK
Extracting  rough/Snow-Covered/Snow (6).jpg
98%  OK
Extracting  rough/Snow-Covered/Snow (7).jpg
98%  OK
Extracting  rough/Snow-Covered/Snow (8).jpg
99%  OK
Extracting  rough/Snow-Covered/Snow (9).jpg
99%  OK
All OK
```

```
image_data='/content/rough'
pd.DataFrame(os.listdir(image_data),columns=['Files_Name'])
```

```
{"summary":"{\n  \"name\": \"pd\",\n  \"rows\": 6,\n  \"fields\": [\n
{\n      \"column\": \"Files_Name\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 6,\n
\"samples\": [\n          \"Dusty\",\n          \"Snow-Covered\",\n
\"Clean\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```

```
img_height = 224
img_width = 224
train_ds = tf.keras.utils.image_dataset_from_directory(
  '/content/rough',
  validation_split=0.2,
  subset='training',
  image_size=(img_height, img_width),
  batch_size=32,
  seed=42,
  shuffle=True)

val_ds = tf.keras.utils.image_dataset_from_directory(
  '/content/rough',
  validation_split=0.2,
  subset='validation',
  image_size=(img_height, img_width),
  batch_size=32,
  seed=42,
  shuffle=True)
```

```
Found 175 files belonging to 6 classes.
Using 140 files for training.
Found 175 files belonging to 6 classes.
Using 35 files for validation.
```

```
# create subdirectories for each class under train and validation
directories
class_names = train_ds.class_names
```

```
for cls in class_names:
    os.makedirs("train/" + cls, exist_ok=True)
    os.makedirs("validation/" + cls, exist_ok=True)

print(class_names)
train_ds
```

```
['Bird-drop', 'Clean', 'Dusty', 'Electrical-damage', 'Physical-
Damage', 'Snow-Covered']
```

```
<_PrefetchDataset element_spec=(TensorSpec(shape=(None, 224, 224, 3),
dtype=tf.float32, name=None), TensorSpec(shape=(None,),
dtype=tf.int32, name=None))>
```

```
import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.applications import ResNet50, Xception,
DenseNet121, VGG16, InceptionV3, EfficientNetB0
from tensorflow.keras.layers import GlobalAveragePooling2D, Dense
from tensorflow.keras.models import Model
from tensorflow.keras.applications import ResNet50
```

ResNet50

```
from tensorflow.keras.applications import ResNet50

# Load the pre-trained ResNet50 model with weights trained on ImageNet
base_model = ResNet50(weights='imagenet', include_top=False)

# Set the trainable attribute for some layers
for layer in base_model.layers[:14]:
    layer.trainable = False

num_classes = 2
# Add a new top layer to the model
x = base_model.output
x = GlobalAveragePooling2D()(x)
output = Dense(num_classes, activation='softmax')(x)

# Create the final model
model = Model(inputs=base_model.input, outputs=output)

# Print the model summary
model.summary()
```

```
Model: "model_8"
______________________________________________________________________
____________________________
 Layer (type)                Output Shape                 Param #
Connected to
```

```
======================================================================
============================
 input_10 (InputLayer)       [(None, None, None, 3)]      0         []



 conv1_pad (ZeroPadding2D)   (None, None, None, 3)        0
['input_10[0][0]']


 conv1_conv (Conv2D)         (None, None, None, 64)       9472
['conv1_pad[0][0]']


 conv1_bn (BatchNormalizati  (None, None, None, 64)       256
['conv1_conv[0][0]']
 on)



 conv1_relu (Activation)     (None, None, None, 64)       0
['conv1_bn[0][0]']


 pool1_pad (ZeroPadding2D)   (None, None, None, 64)       0
['conv1_relu[0][0]']


 pool1_pool (MaxPooling2D)   (None, None, None, 64)       0
['pool1_pad[0][0]']


 conv2_block1_1_conv (Conv2  (None, None, None, 64)       4160
['pool1_pool[0][0]']
 D)



 conv2_block1_1_bn (BatchNo  (None, None, None, 64)       256
['conv2_block1_1_conv[0][0]']
 rmalization)



 conv2_block1_1_relu (Activ  (None, None, None, 64)       0
['conv2_block1_1_bn[0][0]']
 ation)



 conv2_block1_2_conv (Conv2  (None, None, None, 64)       36928
['conv2_block1_1_relu[0][0]']
 D)



 conv2_block1_2_bn (BatchNo  (None, None, None, 64)       256
['conv2_block1_2_conv[0][0]']
 rmalization)



 conv2_block1_2_relu (Activ  (None, None, None, 64)       0
['conv2_block1_2_bn[0][0]']
 ation)



 conv2_block1_0_conv (Conv2  (None, None, None, 256)      16640
['pool1_pool[0][0]']
 D)



 conv2_block1_3_conv (Conv2  (None, None, None, 256)      16640
['conv2_block1_2_relu[0][0]']
 D)



 conv2_block1_0_bn (BatchNo  (None, None, None, 256)      1024
['conv2_block1_0_conv[0][0]']
 rmalization)



 conv2_block1_3_bn (BatchNo  (None, None, None, 256)      1024
['conv2_block1_3_conv[0][0]']
 rmalization)



 conv2_block1_add (Add)      (None, None, None, 256)      0
['conv2_block1_0_bn[0][0]',

'conv2_block1_3_bn[0][0]']


 conv2_block1_out (Activati  (None, None, None, 256)      0
['conv2_block1_add[0][0]']
 on)
 conv2_block2_1_conv (Conv2  (None, None, None, 64)       16448
['conv2_block1_out[0][0]']
 D)



 conv2_block2_1_bn (BatchNo  (None, None, None, 64)       256
['conv2_block2_1_conv[0][0]']
 rmalization)



 conv2_block2_1_relu (Activ  (None, None, None, 64)       0
['conv2_block2_1_bn[0][0]']
 ation)



 conv2_block2_2_conv (Conv2  (None, None, None, 64)       36928
['conv2_block2_1_relu[0][0]']
 D)



 conv2_block2_2_bn (BatchNo  (None, None, None, 64)       256
['conv2_block2_2_conv[0][0]']
 rmalization)



 conv2_block2_2_relu (Activ  (None, None, None, 64)       0
['conv2_block2_2_bn[0][0]']
 ation)



 conv2_block2_3_conv (Conv2  (None, None, None, 256)      16640
['conv2_block2_2_relu[0][0]']
 D)



 conv2_block2_3_bn (BatchNo  (None, None, None, 256)      1024
['conv2_block2_3_conv[0][0]']
 rmalization)



 conv2_block2_add (Add)      (None, None, None, 256)      0
```

```
['conv2_block1_out[0][0]',

'conv2_block2_3_bn[0][0]']


 conv2_block2_out (Activati  (None, None, None, 256)      0
['conv2_block2_add[0][0]']
 on)



 conv2_block3_1_conv (Conv2  (None, None, None, 64)       16448
['conv2_block2_out[0][0]']
 D)



 conv2_block3_1_bn (BatchNo  (None, None, None, 64)       256
['conv2_block3_1_conv[0][0]']
 rmalization)



 conv2_block3_1_relu (Activ  (None, None, None, 64)       0
['conv2_block3_1_bn[0][0]']
 ation)



 conv2_block3_2_conv (Conv2  (None, None, None, 64)       36928
['conv2_block3_1_relu[0][0]']
 D)



 conv2_block3_2_bn (BatchNo  (None, None, None, 64)       256
['conv2_block3_2_conv[0][0]']
 rmalization)



 conv2_block3_2_relu (Activ  (None, None, None, 64)       0
['conv2_block3_2_bn[0][0]']
 ation)



 conv2_block3_3_conv (Conv2  (None, None, None, 256)      16640
['conv2_block3_2_relu[0][0]']
 D)
 conv2_block3_3_bn (BatchNo  (None, None, None, 256)      1024
['conv2_block3_3_conv[0][0]']
 rmalization)


 conv2_block3_add (Add)      (None, None, None, 256)      0
['conv2_block2_out[0][0]',

'conv2_block3_3_bn[0][0]']


 conv2_block3_out (Activati  (None, None, None, 256)      0
['conv2_block3_add[0][0]']
 on)


 conv3_block1_1_conv (Conv2  (None, None, None, 128)      32896
['conv2_block3_out[0][0]']
 D)


 conv3_block1_1_bn (BatchNo  (None, None, None, 128)      512
['conv3_block1_1_conv[0][0]']
 rmalization)


 conv3_block1_1_relu (Activ  (None, None, None, 128)      0
['conv3_block1_1_bn[0][0]']
 ation)


 conv3_block1_2_conv (Conv2  (None, None, None, 128)      147584
['conv3_block1_1_relu[0][0]']
 D)


 conv3_block1_2_bn (BatchNo  (None, None, None, 128)      512
['conv3_block1_2_conv[0][0]']
 rmalization)


 conv3_block1_2_relu (Activ  (None, None, None, 128)      0
['conv3_block1_2_bn[0][0]']
 ation)



 conv3_block1_0_conv (Conv2  (None, None, None, 512)      131584
['conv2_block3_out[0][0]']
 D)



 conv3_block1_3_conv (Conv2  (None, None, None, 512)      66048
['conv3_block1_2_relu[0][0]']
 D)



 conv3_block1_0_bn (BatchNo  (None, None, None, 512)      2048
['conv3_block1_0_conv[0][0]']
 rmalization)



 conv3_block1_3_bn (BatchNo  (None, None, None, 512)      2048
['conv3_block1_3_conv[0][0]']
 rmalization)



 conv3_block1_add (Add)      (None, None, None, 512)      0
['conv3_block1_0_bn[0][0]',

'conv3_block1_3_bn[0][0]']


 conv3_block1_out (Activati  (None, None, None, 512)      0
['conv3_block1_add[0][0]']
 on)



 conv3_block2_1_conv (Conv2  (None, None, None, 128)      65664
['conv3_block1_out[0][0]']
 D)



 conv3_block2_1_bn (BatchNo  (None, None, None, 128)      512
['conv3_block2_1_conv[0][0]']
 rmalization)
 conv3_block2_1_relu (Activ  (None, None, None, 128)      0
['conv3_block2_1_bn[0][0]']
 ation)



 conv3_block2_2_conv (Conv2  (None, None, None, 128)      147584
['conv3_block2_1_relu[0][0]']
 D)



 conv3_block2_2_bn (BatchNo  (None, None, None, 128)      512
['conv3_block2_2_conv[0][0]']
 rmalization)



 conv3_block2_2_relu (Activ  (None, None, None, 128)      0
['conv3_block2_2_bn[0][0]']
 ation)



 conv3_block2_3_conv (Conv2  (None, None, None, 512)      66048
['conv3_block2_2_relu[0][0]']
 D)



 conv3_block2_3_bn (BatchNo  (None, None, None, 512)      2048
['conv3_block2_3_conv[0][0]']
 rmalization)



 conv3_block2_add (Add)      (None, None, None, 512)      0
['conv3_block1_out[0][0]',

'conv3_block2_3_bn[0][0]']


 conv3_block2_out (Activati  (None, None, None, 512)      0
['conv3_block2_add[0][0]']
 on)



 conv3_block3_1_conv (Conv2  (None, None, None, 128)      65664
['conv3_block2_out[0][0]']
 D)



 conv3_block3_1_bn (BatchNo  (None, None, None, 128)      512
['conv3_block3_1_conv[0][0]']
 rmalization)



 conv3_block3_1_relu (Activ  (None, None, None, 128)      0
['conv3_block3_1_bn[0][0]']
 ation)



 conv3_block3_2_conv (Conv2  (None, None, None, 128)      147584
['conv3_block3_1_relu[0][0]']
 D)



 conv3_block3_2_bn (BatchNo  (None, None, None, 128)      512
['conv3_block3_2_conv[0][0]']
 rmalization)



 conv3_block3_2_relu (Activ  (None, None, None, 128)      0
['conv3_block3_2_bn[0][0]']
 ation)



 conv3_block3_3_conv (Conv2  (None, None, None, 512)      66048
['conv3_block3_2_relu[0][0]']
 D)



 conv3_block3_3_bn (BatchNo  (None, None, None, 512)      2048
['conv3_block3_3_conv[0][0]']
 rmalization)



 conv3_block3_add (Add)      (None, None, None, 512)      0
['conv3_block2_out[0][0]',

'conv3_block3_3_bn[0][0]']
```

```
 conv3_block3_out (Activati  (None, None, None, 512)      0
['conv3_block3_add[0][0]']
 on)


 conv3_block4_1_conv (Conv2  (None, None, None, 128)      65664
['conv3_block3_out[0][0]']
 D)


 conv3_block4_1_bn (BatchNo  (None, None, None, 128)      512
['conv3_block4_1_conv[0][0]']
 rmalization)


 conv3_block4_1_relu (Activ  (None, None, None, 128)      0
['conv3_block4_1_bn[0][0]']
 ation)


 conv3_block4_2_conv (Conv2  (None, None, None, 128)      147584
['conv3_block4_1_relu[0][0]']
 D)


 conv3_block4_2_bn (BatchNo  (None, None, None, 128)      512
['conv3_block4_2_conv[0][0]']
 rmalization)


 conv3_block4_2_relu (Activ  (None, None, None, 128)      0
['conv3_block4_2_bn[0][0]']
 ation)


 conv3_block4_3_conv (Conv2  (None, None, None, 512)      66048
['conv3_block4_2_relu[0][0]']
 D)


 conv3_block4_3_bn (BatchNo  (None, None, None, 512)      2048
['conv3_block4_3_conv[0][0]']
 rmalization)



 conv3_block4_add (Add)      (None, None, None, 512)      0
['conv3_block3_out[0][0]',

'conv3_block4_3_bn[0][0]']


 conv3_block4_out (Activati  (None, None, None, 512)      0
['conv3_block4_add[0][0]']
 on)



 conv4_block1_1_conv (Conv2  (None, None, None, 256)      131328
['conv3_block4_out[0][0]']
 D)



 conv4_block1_1_bn (BatchNo  (None, None, None, 256)      1024
['conv4_block1_1_conv[0][0]']
 rmalization)



 conv4_block1_1_relu (Activ  (None, None, None, 256)      0
['conv4_block1_1_bn[0][0]']
 ation)



 conv4_block1_2_conv (Conv2  (None, None, None, 256)      590080
['conv4_block1_1_relu[0][0]']
 D)



 conv4_block1_2_bn (BatchNo  (None, None, None, 256)      1024
['conv4_block1_2_conv[0][0]']
 rmalization)



 conv4_block1_2_relu (Activ  (None, None, None, 256)      0
['conv4_block1_2_bn[0][0]']
 ation)
 conv4_block1_0_conv (Conv2  (None, None, None, 1024)     525312
['conv3_block4_out[0][0]']
 D)



 conv4_block1_3_conv (Conv2  (None, None, None, 1024)     263168
['conv4_block1_2_relu[0][0]']
 D)



 conv4_block1_0_bn (BatchNo  (None, None, None, 1024)     4096
['conv4_block1_0_conv[0][0]']
 rmalization)



 conv4_block1_3_bn (BatchNo  (None, None, None, 1024)     4096
['conv4_block1_3_conv[0][0]']
 rmalization)



 conv4_block1_add (Add)      (None, None, None, 1024)     0
['conv4_block1_0_bn[0][0]',

'conv4_block1_3_bn[0][0]']


 conv4_block1_out (Activati  (None, None, None, 1024)     0
['conv4_block1_add[0][0]']
 on)



 conv4_block2_1_conv (Conv2  (None, None, None, 256)      262400
['conv4_block1_out[0][0]']
 D)



 conv4_block2_1_bn (BatchNo  (None, None, None, 256)      1024
['conv4_block2_1_conv[0][0]']
 rmalization)



 conv4_block2_1_relu (Activ  (None, None, None, 256)      0
['conv4_block2_1_bn[0][0]']
 ation)



 conv4_block2_2_conv (Conv2  (None, None, None, 256)      590080
['conv4_block2_1_relu[0][0]']
 D)



 conv4_block2_2_bn (BatchNo  (None, None, None, 256)      1024
['conv4_block2_2_conv[0][0]']
 rmalization)



 conv4_block2_2_relu (Activ  (None, None, None, 256)      0
['conv4_block2_2_bn[0][0]']
 ation)



 conv4_block2_3_conv (Conv2  (None, None, None, 1024)     263168
['conv4_block2_2_relu[0][0]']
 D)



 conv4_block2_3_bn (BatchNo  (None, None, None, 1024)     4096
['conv4_block2_3_conv[0][0]']
 rmalization)



 conv4_block2_add (Add)      (None, None, None, 1024)     0
['conv4_block1_out[0][0]',

'conv4_block2_3_bn[0][0]']


 conv4_block2_out (Activati  (None, None, None, 1024)     0
['conv4_block2_add[0][0]']
 on)



 conv4_block3_1_conv (Conv2  (None, None, None, 256)      262400
['conv4_block2_out[0][0]']
 D)
 conv4_block3_1_bn (BatchNo  (None, None, None, 256)      1024
['conv4_block3_1_conv[0][0]']
 rmalization)



 conv4_block3_1_relu (Activ  (None, None, None, 256)      0
['conv4_block3_1_bn[0][0]']
 ation)



 conv4_block3_2_conv (Conv2  (None, None, None, 256)      590080
['conv4_block3_1_relu[0][0]']
 D)



 conv4_block3_2_bn (BatchNo  (None, None, None, 256)      1024
['conv4_block3_2_conv[0][0]']
 rmalization)



 conv4_block3_2_relu (Activ  (None, None, None, 256)      0
['conv4_block3_2_bn[0][0]']
 ation)



 conv4_block3_3_conv (Conv2  (None, None, None, 1024)     263168
['conv4_block3_2_relu[0][0]']
 D)



 conv4_block3_3_bn (BatchNo  (None, None, None, 1024)     4096
['conv4_block3_3_conv[0][0]']
 rmalization)



 conv4_block3_add (Add)      (None, None, None, 1024)     0
['conv4_block2_out[0][0]',

'conv4_block3_3_bn[0][0]']


 conv4_block3_out (Activati  (None, None, None, 1024)     0
['conv4_block3_add[0][0]']
 on)



 conv4_block4_1_conv (Conv2  (None, None, None, 256)      262400
['conv4_block3_out[0][0]']
 D)



 conv4_block4_1_bn (BatchNo  (None, None, None, 256)      1024
['conv4_block4_1_conv[0][0]']
 rmalization)



 conv4_block4_1_relu (Activ  (None, None, None, 256)      0
['conv4_block4_1_bn[0][0]']
 ation)



 conv4_block4_2_conv (Conv2  (None, None, None, 256)      590080
['conv4_block4_1_relu[0][0]']
 D)



 conv4_block4_2_bn (BatchNo  (None, None, None, 256)      1024
['conv4_block4_2_conv[0][0]']
 rmalization)



 conv4_block4_2_relu (Activ  (None, None, None, 256)      0
['conv4_block4_2_bn[0][0]']
 ation)



 conv4_block4_3_conv (Conv2  (None, None, None, 1024)     263168
['conv4_block4_2_relu[0][0]']
 D)



 conv4_block4_3_bn (BatchNo  (None, None, None, 1024)     4096
['conv4_block4_3_conv[0][0]']
 rmalization)
 conv4_block4_add (Add)       (None, None, None, 1024)     0
['conv4_block3_out[0][0]',

'conv4_block4_3_bn[0][0]']


 conv4_block4_out (Activati  (None, None, None, 1024)     0
['conv4_block4_add[0][0]']
 on)



 conv4_block5_1_conv (Conv2  (None, None, None, 256)      262400
['conv4_block4_out[0][0]']
 D)



 conv4_block5_1_bn (BatchNo  (None, None, None, 256)      1024
['conv4_block5_1_conv[0][0]']
 rmalization)



 conv4_block5_1_relu (Activ  (None, None, None, 256)      0
['conv4_block5_1_bn[0][0]']
 ation)



 conv4_block5_2_conv (Conv2  (None, None, None, 256)      590080
['conv4_block5_1_relu[0][0]']
 D)



 conv4_block5_2_bn (BatchNo  (None, None, None, 256)      1024
['conv4_block5_2_conv[0][0]']
 rmalization)



 conv4_block5_2_relu (Activ  (None, None, None, 256)      0
['conv4_block5_2_bn[0][0]']
 ation)



 conv4_block5_3_conv (Conv2  (None, None, None, 1024)     263168
['conv4_block5_2_relu[0][0]']
 D)



 conv4_block5_3_bn (BatchNo  (None, None, None, 1024)     4096
['conv4_block5_3_conv[0][0]']
 rmalization)



 conv4_block5_add (Add)      (None, None, None, 1024)     0
['conv4_block4_out[0][0]',

'conv4_block5_3_bn[0][0]']


 conv4_block5_out (Activati  (None, None, None, 1024)     0
['conv4_block5_add[0][0]']
 on)



 conv4_block6_1_conv (Conv2  (None, None, None, 256)      262400
['conv4_block5_out[0][0]']
 D)



 conv4_block6_1_bn (BatchNo  (None, None, None, 256)      1024
['conv4_block6_1_conv[0][0]']
 rmalization)



 conv4_block6_1_relu (Activ  (None, None, None, 256)      0
['conv4_block6_1_bn[0][0]']
 ation)



 conv4_block6_2_conv (Conv2  (None, None, None, 256)      590080
['conv4_block6_1_relu[0][0]']
 D)



 conv4_block6_2_bn (BatchNo  (None, None, None, 256)      1024
['conv4_block6_2_conv[0][0]']
 rmalization)
 conv4_block6_2_relu (Activ  (None, None, None, 256)      0
['conv4_block6_2_bn[0][0]']
 ation)



 conv4_block6_3_conv (Conv2  (None, None, None, 1024)     263168
['conv4_block6_2_relu[0][0]']
 D)



 conv4_block6_3_bn (BatchNo  (None, None, None, 1024)     4096
['conv4_block6_3_conv[0][0]']
 rmalization)



 conv4_block6_add (Add)      (None, None, None, 1024)     0
['conv4_block5_out[0][0]',

'conv4_block6_3_bn[0][0]']


 conv4_block6_out (Activati  (None, None, None, 1024)     0
['conv4_block6_add[0][0]']
 on)



 conv5_block1_1_conv (Conv2  (None, None, None, 512)      524800
['conv4_block6_out[0][0]']
 D)



 conv5_block1_1_bn (BatchNo  (None, None, None, 512)      2048
['conv5_block1_1_conv[0][0]']
 rmalization)



 conv5_block1_1_relu (Activ  (None, None, None, 512)      0
['conv5_block1_1_bn[0][0]']
 ation)



 conv5_block1_2_conv (Conv2  (None, None, None, 512)      2359808
['conv5_block1_1_relu[0][0]']
 D)



 conv5_block1_2_bn (BatchNo  (None, None, None, 512)      2048
['conv5_block1_2_conv[0][0]']
 rmalization)



 conv5_block1_2_relu (Activ  (None, None, None, 512)      0
['conv5_block1_2_bn[0][0]']
 ation)



 conv5_block1_0_conv (Conv2  (None, None, None, 2048)     2099200
['conv4_block6_out[0][0]']
 D)



 conv5_block1_3_conv (Conv2  (None, None, None, 2048)     1050624
['conv5_block1_2_relu[0][0]']
 D)



 conv5_block1_0_bn (BatchNo  (None, None, None, 2048)     8192
['conv5_block1_0_conv[0][0]']
 rmalization)



 conv5_block1_3_bn (BatchNo  (None, None, None, 2048)     8192
['conv5_block1_3_conv[0][0]']
 rmalization)



 conv5_block1_add (Add)      (None, None, None, 2048)     0
['conv5_block1_0_bn[0][0]',

'conv5_block1_3_bn[0][0]']


 conv5_block1_out (Activati  (None, None, None, 2048)     0
['conv5_block1_add[0][0]']
 on)
 conv5_block2_1_conv (Conv2  (None, None, None, 512)      1049088
['conv5_block1_out[0][0]']
 D)



 conv5_block2_1_bn (BatchNo  (None, None, None, 512)      2048
['conv5_block2_1_conv[0][0]']
 rmalization)



 conv5_block2_1_relu (Activ  (None, None, None, 512)      0
['conv5_block2_1_bn[0][0]']
 ation)



 conv5_block2_2_conv (Conv2  (None, None, None, 512)      2359808
['conv5_block2_1_relu[0][0]']
 D)



 conv5_block2_2_bn (BatchNo  (None, None, None, 512)      2048
['conv5_block2_2_conv[0][0]']
 rmalization)



 conv5_block2_2_relu (Activ  (None, None, None, 512)      0
['conv5_block2_2_bn[0][0]']
 ation)



 conv5_block2_3_conv (Conv2  (None, None, None, 2048)     1050624
['conv5_block2_2_relu[0][0]']
 D)



 conv5_block2_3_bn (BatchNo  (None, None, None, 2048)     8192
['conv5_block2_3_conv[0][0]']
 rmalization)



 conv5_block2_add (Add)      (None, None, None, 2048)     0
```

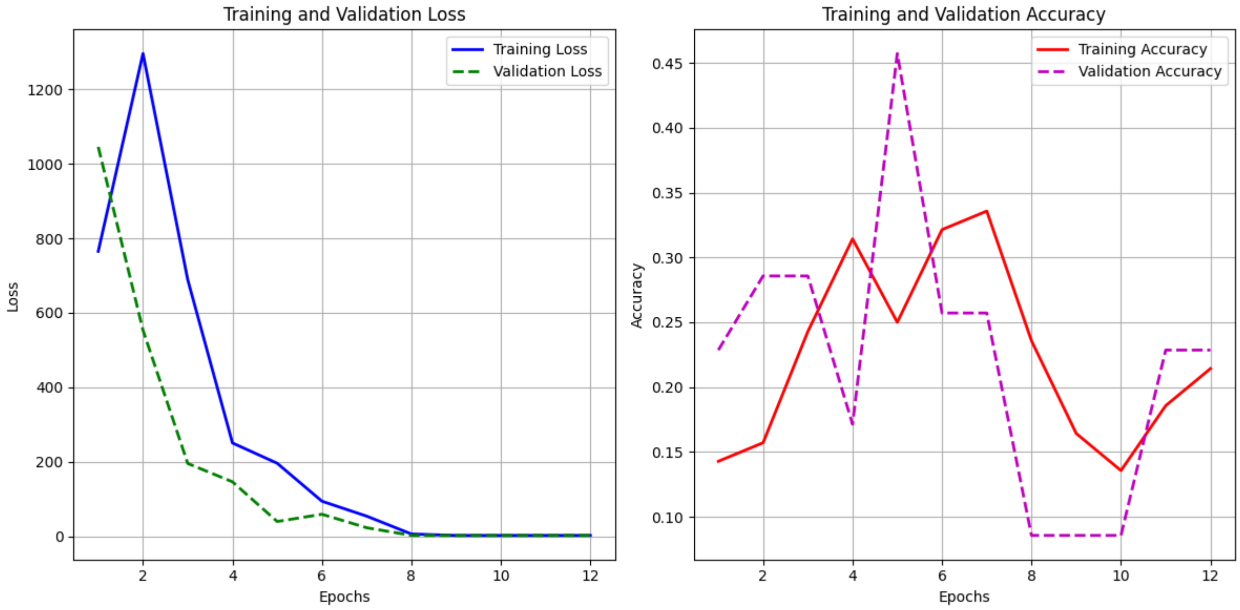
```
['conv5_block1_out[0][0]',

'conv5_block2_3_bn[0][0]']


 conv5_block2_out (Activati  (None, None, None, 2048)     0
['conv5_block2_add[0][0]']
 on)



 conv5_block3_1_conv (Conv2  (None, None, None, 512)      1049088
['conv5_block2_out[0][0]']
 D)



 conv5_block3_1_bn (BatchNo  (None, None, None, 512)      2048
['conv5_block3_1_conv[0][0]']
 rmalization)



 conv5_block3_1_relu (Activ  (None, None, None, 512)      0
['conv5_block3_1_bn[0][0]']
 ation)



 conv5_block3_2_conv (Conv2  (None, None, None, 512)      2359808
['conv5_block3_1_relu[0][0]']
 D)



 conv5_block3_2_bn (BatchNo  (None, None, None, 512)      2048
['conv5_block3_2_conv[0][0]']
 rmalization)



 conv5_block3_2_relu (Activ  (None, None, None, 512)      0
['conv5_block3_2_bn[0][0]']
 ation)



 conv5_block3_3_conv (Conv2  (None, None, None, 2048)     1050624
['conv5_block3_2_relu[0][0]']
 D)
 conv5_block3_3_bn (BatchNo  (None, None, None, 2048)     8192
['conv5_block3_3_conv[0][0]']
 rmalization)



 conv5_block3_add (Add)      (None, None, None, 2048)     0
['conv5_block2_out[0][0]',

'conv5_block3_3_bn[0][0]']


 conv5_block3_out (Activati  (None, None, None, 2048)     0
['conv5_block3_add[0][0]']
 on)



 global_average_pooling2d_6  (None, 2048)                 0
['conv5_block3_out[0][0]']
  (GlobalAveragePooling2D)



 dense_5 (Dense)             (None, 2)                    4098
['global_average_pooling2d_6[0
                                                                    ]
[0]']


======================================================================
============================
Total params: 23591810 (90.00 MB)
Trainable params: 23471106 (89.54 MB)
Non-trainable params: 120704 (471.50 KB)
______________________________________________________________________
____________________________

tf.keras.layers.Conv2D(32, (3, 3), activation='relu',
input_shape=(224, 224, 3)),

model.compile(optimizer=tf.keras.optimizers.Adam(0.0001),

loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu',
input_shape=(224, 224, 3)),
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation='softmax')
])

model.compile(optimizer=tf.keras.optimizers.Adam(0.0001),

loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

epoch = 15
history = model.fit(train_ds, validation_data=val_ds, epochs=epoch,
    callbacks = [
        tf.keras.callbacks.EarlyStopping(
            monitor="val_loss",
            min_delta=1e-2,
            patience=3,
            verbose=1,
        )
    ]
)

Epoch 1/15
5/5 [==============================] - 15s 2s/step - loss: 765.1240 -
accuracy: 0.1429 - val_loss: 1045.4414 - val_accuracy: 0.2286
Epoch 2/15
5/5 [==============================] - 14s 3s/step - loss: 1296.0035 -
accuracy: 0.1571 - val_loss: 553.2461 - val_accuracy: 0.2857
Epoch 3/15
5/5 [==============================] - 13s 2s/step - loss: 688.7361 -
accuracy: 0.2429 - val_loss: 195.4565 - val_accuracy: 0.2857
Epoch 4/15
5/5 [==============================] - 13s 2s/step - loss: 250.4739 -
accuracy: 0.3143 - val_loss: 146.1383 - val_accuracy: 0.1714
Epoch 5/15
5/5 [==============================] - 13s 2s/step - loss: 195.9534 -
accuracy: 0.2500 - val_loss: 39.5370 - val_accuracy: 0.4571
Epoch 6/15
5/5 [==============================] - 13s 2s/step - loss: 94.0472 -
accuracy: 0.3214 - val_loss: 59.0161 - val_accuracy: 0.2571
Epoch 7/15
5/5 [==============================] - 13s 2s/step - loss: 53.9503 -
accuracy: 0.3357 - val_loss: 22.8814 - val_accuracy: 0.2571
Epoch 8/15
5/5 [==============================] - 13s 2s/step - loss: 6.0462 -
accuracy: 0.2357 - val_loss: 2.3930 - val_accuracy: 0.0857
Epoch 9/15
5/5 [==============================] - 12s 2s/step - loss: 2.2303 -
```

```
accuracy: 0.1643 - val_loss: 2.3019 - val_accuracy: 0.0857
Epoch 10/15
5/5 [==============================] - 12s 2s/step - loss: 2.2840 -
accuracy: 0.1357 - val_loss: 2.3016 - val_accuracy: 0.0857
Epoch 11/15
5/5 [==============================] - 14s 2s/step - loss: 2.2850 -
accuracy: 0.1857 - val_loss: 2.3012 - val_accuracy: 0.2286
Epoch 12/15
5/5 [==============================] - 13s 2s/step - loss: 2.2846 -
accuracy: 0.2143 - val_loss: 2.3009 - val_accuracy: 0.2286
Epoch 12: early stopping
```

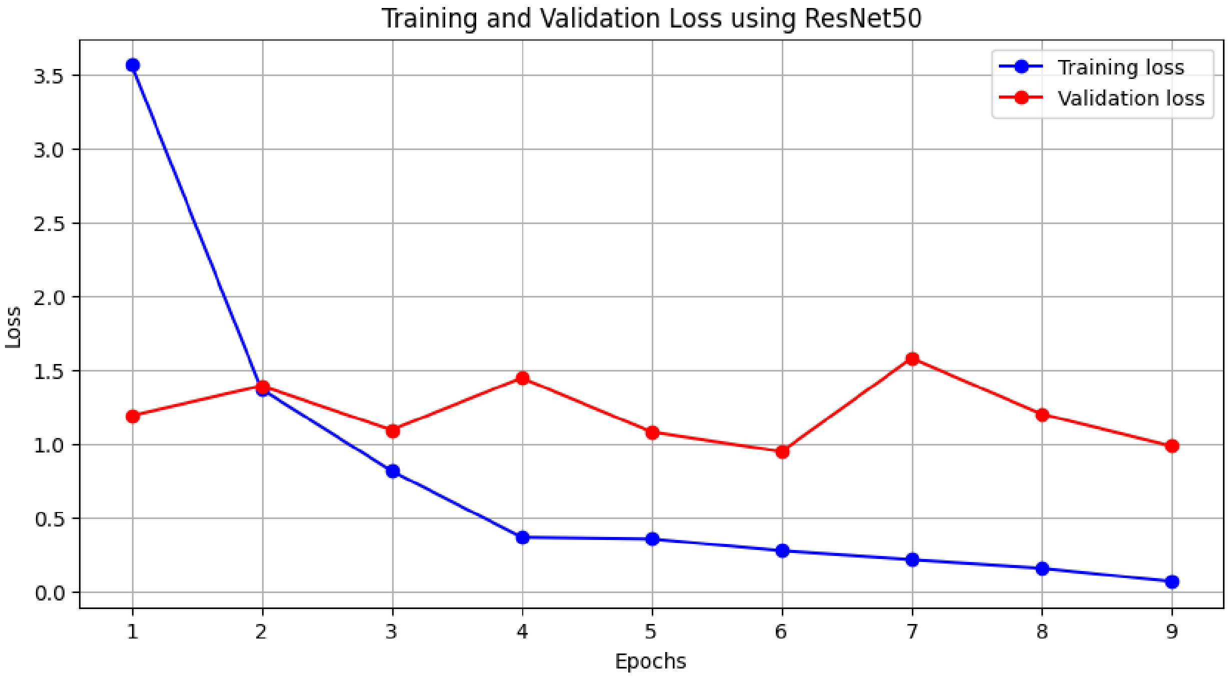
```
import numpy as np

predictions = model.predict(train_ds)
damage_masks = predictions > 0.5
damage_percentages = [np.mean(mask) * 100 for mask in damage_masks]
average_damage_percentage = np.mean(damage_percentages)
print(f"Average Damage Percentage: {average_damage_percentage:.2f}%")
```

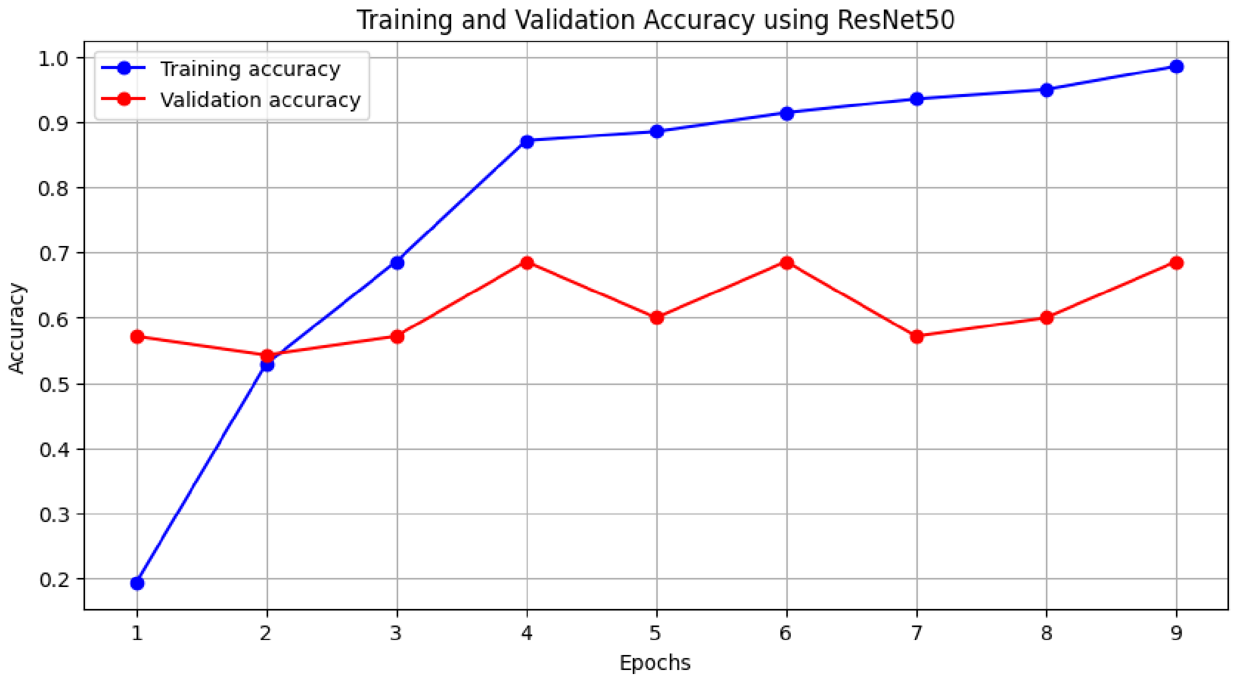
```
5/5 [==============================] - 5s 349ms/step
Average Damage Percentage: 0.07%
```

```
import matplotlib.pyplot as plt

# Extract the history from the training
loss = history.history['loss']
val_loss = history.history['val_loss']
accuracy = history.history.get('accuracy', None)  # Use get in case
accuracy was not recorded
val_accuracy = history.history.get('val_accuracy', None)

# Number of epochs actually run
epochs = range(1, len(loss) + 1)

# Create a plot for the loss
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.plot(epochs, loss, 'b-', label='Training Loss', linewidth=2)
plt.plot(epochs, val_loss, 'g--', label='Validation Loss',
linewidth=2)
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.grid(True)

# Create a plot for the accuracy if recorded
if accuracy is not None and val_accuracy is not None:
    plt.subplot(1, 2, 2)
    plt.plot(epochs, accuracy, 'r-', label='Training Accuracy',
linewidth=2)
    plt.plot(epochs, val_accuracy, 'm--', label='Validation Accuracy',
linewidth=2)
    plt.title('Training and Validation Accuracy')
    plt.xlabel('Epochs')
    plt.ylabel('Accuracy')
    plt.legend()
    plt.grid(True)

plt.tight_layout()
plt.show()
```



```
import matplotlib.pyplot as plt

# Define the epoch data
epochs = [1, 2, 3, 4, 5, 6, 7, 8, 9]
train_loss = [3.5652, 1.3672, 0.8142, 0.3656, 0.3535, 0.2729, 0.2131,
0.1550, 0.0651]
train_accuracy = [0.1929, 0.5286, 0.6857, 0.8714, 0.8857, 0.9143,
0.9357, 0.9500, 0.9857]
val_loss = [1.1886, 1.3906, 1.0916, 1.4441, 1.0783, 0.9454, 1.5770,
1.1991, 0.9830]
val_accuracy = [0.5714, 0.5429, 0.5714, 0.6857, 0.6000, 0.6857,
0.5714, 0.6000, 0.6857]

# Plot training and validation loss
plt.figure(figsize=(10, 5))
plt.plot(epochs, train_loss, 'b-o', label='Training loss')
plt.plot(epochs, val_loss, 'r-o', label='Validation loss')
plt.title('Training and Validation Loss using ResNet50')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.grid(True)
plt.show()

# Plot training and validation accuracy
plt.figure(figsize=(10, 5))
plt.plot(epochs, train_accuracy, 'b-o', label='Training accuracy')
plt.plot(epochs, val_accuracy, 'r-o', label='Validation accuracy')
plt.title('Training and Validation Accuracy using ResNet50')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.grid(True)
plt.show()
```





CNN

```
cnn = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu',
input_shape=(224, 224, 3)),
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation='softmax')
])

cnn.compile(optimizer=tf.keras.optimizers.Adam(0.0001),

loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

history = cnn.fit(train_ds, validation_data=val_ds, epochs=15,
    callbacks = [
        tf.keras.callbacks.EarlyStopping(
            monitor="val_loss",
            min_delta=1e-2,
            patience=3,
            verbose=1,
        )
    ]
)
```

```
Epoch 1/15
```

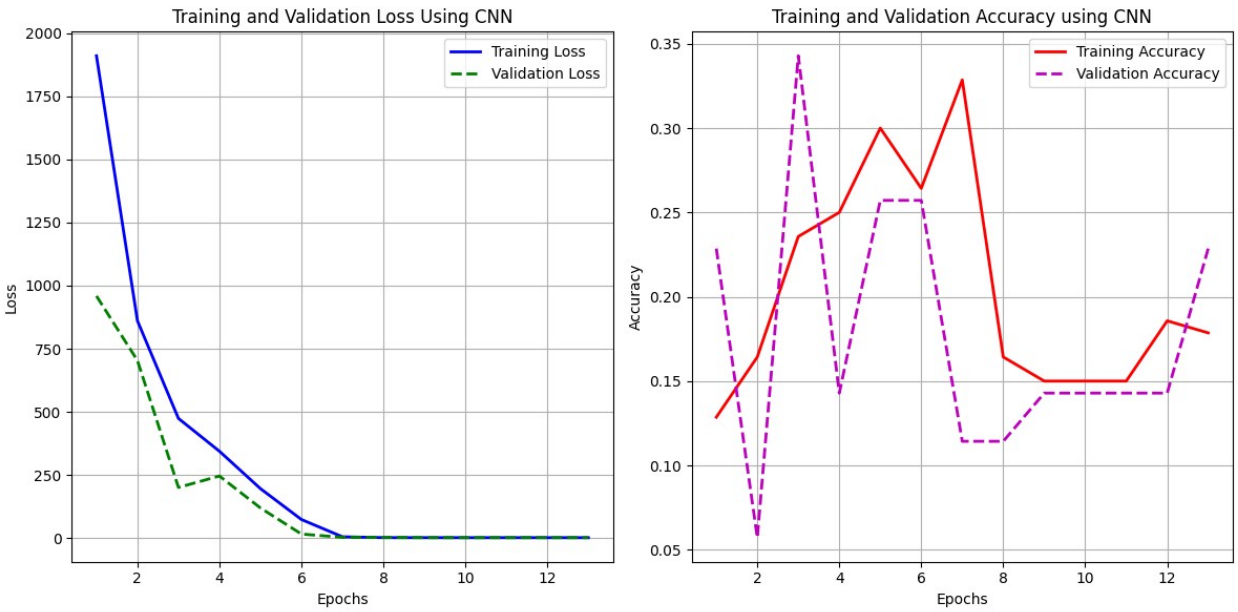
```
/usr/local/lib/python3.10/dist-packages/keras/src/backend.py:5727:
UserWarning: "`sparse_categorical_crossentropy` received
`from_logits=True`, but the `output` argument was produced by a
Softmax activation and thus does not represent logits. Was this
intended?
  output, from_logits = _get_logits(

5/5 [==============================] - 16s 3s/step - loss: 1909.5582 -
accuracy: 0.1286 - val_loss: 958.7530 - val_accuracy: 0.2286
Epoch 2/15
5/5 [==============================] - 13s 2s/step - loss: 860.4562 -
accuracy: 0.1643 - val_loss: 703.2527 - val_accuracy: 0.0571
Epoch 3/15
5/5 [==============================] - 13s 2s/step - loss: 474.1021 -
accuracy: 0.2357 - val_loss: 201.0027 - val_accuracy: 0.3429
Epoch 4/15
5/5 [==============================] - 12s 2s/step - loss: 344.2424 -
accuracy: 0.2500 - val_loss: 246.1908 - val_accuracy: 0.1429
Epoch 5/15
5/5 [==============================] - 13s 2s/step - loss: 196.3222 -
accuracy: 0.3000 - val_loss: 119.7609 - val_accuracy: 0.2571
Epoch 6/15
5/5 [==============================] - 13s 2s/step - loss: 74.0629 -
accuracy: 0.2643 - val_loss: 16.4834 - val_accuracy: 0.2571
Epoch 7/15
5/5 [==============================] - 13s 2s/step - loss: 5.6263 -
accuracy: 0.3286 - val_loss: 3.4625 - val_accuracy: 0.1143
Epoch 8/15
5/5 [==============================] - 12s 2s/step - loss: 2.3023 -
accuracy: 0.1643 - val_loss: 2.9624 - val_accuracy: 0.1143
Epoch 9/15
5/5 [==============================] - 12s 2s/step - loss: 2.2625 -
accuracy: 0.1500 - val_loss: 2.2515 - val_accuracy: 0.1429
Epoch 10/15
5/5 [==============================] - 12s 2s/step - loss: 2.1543 -
accuracy: 0.1500 - val_loss: 2.1899 - val_accuracy: 0.1429
Epoch 11/15
5/5 [==============================] - 14s 2s/step - loss: 2.2405 -
accuracy: 0.1500 - val_loss: 2.1897 - val_accuracy: 0.1429
Epoch 12/15
5/5 [==============================] - 13s 2s/step - loss: 2.1819 -
accuracy: 0.1857 - val_loss: 2.1800 - val_accuracy: 0.1429
Epoch 13/15
5/5 [==============================] - 12s 2s/step - loss: 2.1948 -
accuracy: 0.1786 - val_loss: 2.1890 - val_accuracy: 0.2286
Epoch 13: early stopping
```

```python
import numpy as np

predictions = cnn.predict(train_ds)
```

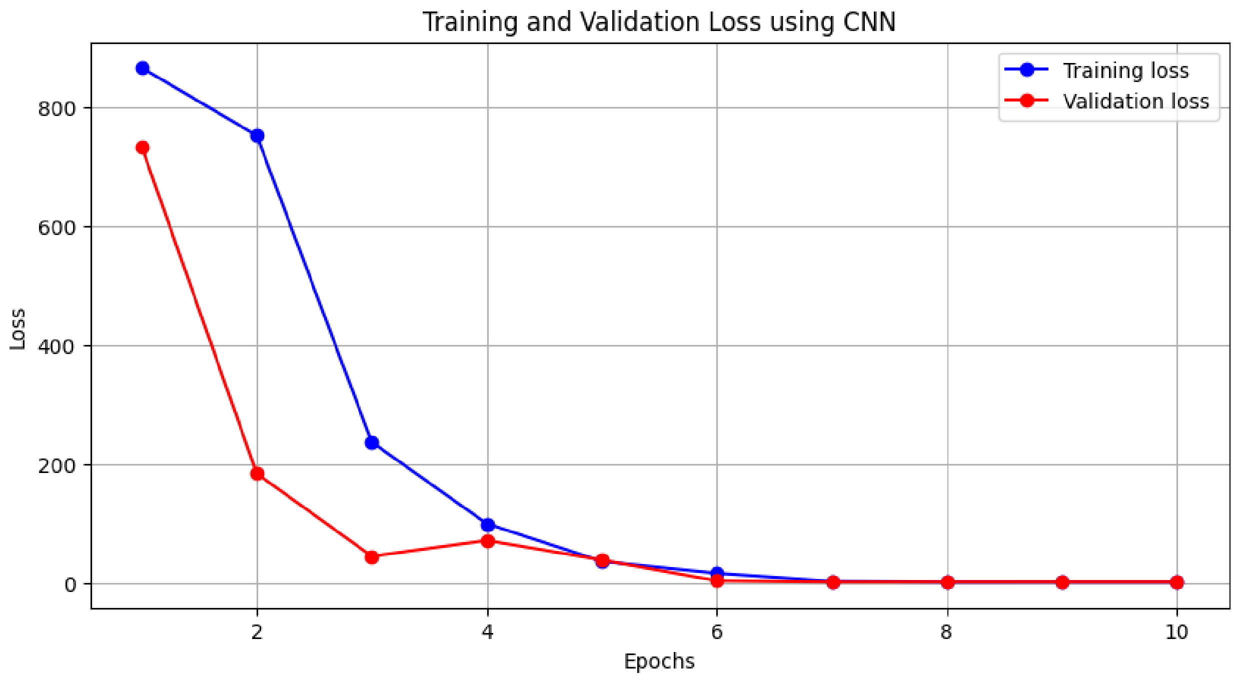
```
damage_masks = predictions > 0.5
damage_percentages = [np.mean(mask) * 100 for mask in damage_masks]
average_damage_percentage = np.mean(damage_percentages)
print(f"Average Damage Percentage: {average_damage_percentage:.2f}%")
```

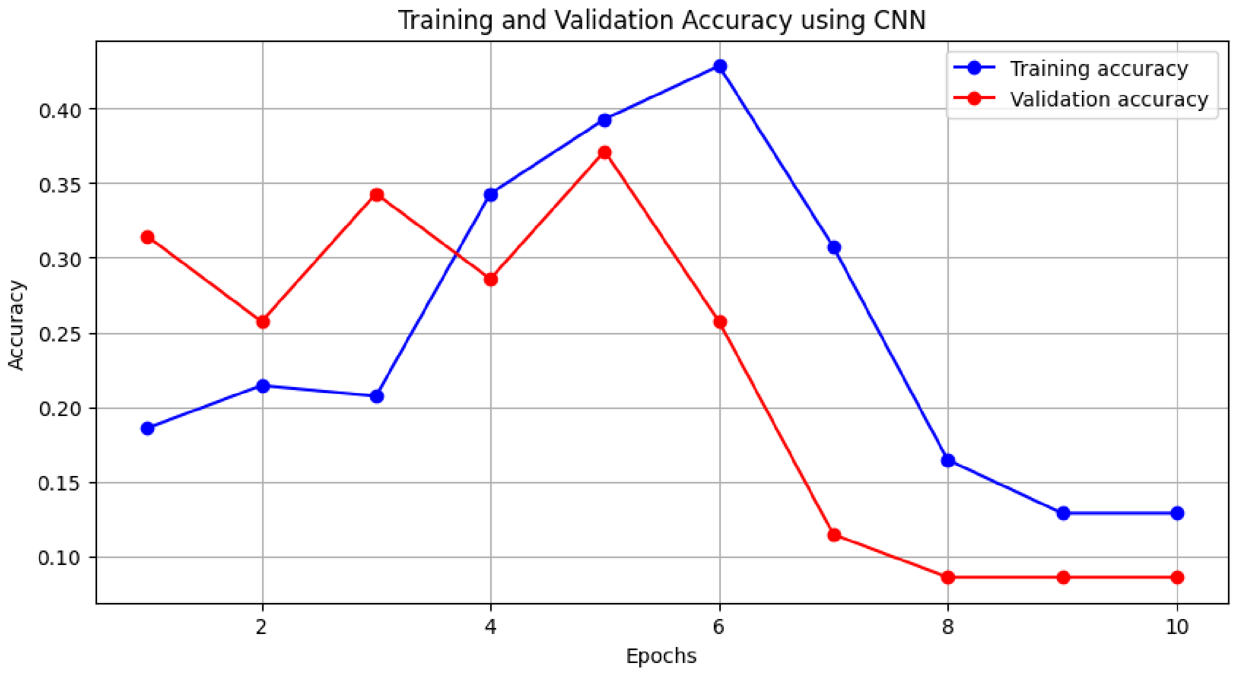
```
5/5 [==============================] - 26s 338ms/step
Average Damage Percentage: 0.93%
```

```
# Extract the history from the training
loss = history.history['loss']
val_loss = history.history['val_loss']
accuracy = history.history.get('accuracy', None)  # Use get in case
accuracy was not recorded
val_accuracy = history.history.get('val_accuracy', None)

# Number of epochs actually run
epochs = range(1, len(loss) + 1)

# Create a plot for the loss
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.plot(epochs, loss, 'b-', label='Training Loss', linewidth=2)
plt.plot(epochs, val_loss, 'g--', label='Validation Loss',
linewidth=2)
plt.title('Training and Validation Loss Using CNN')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.grid(True)

# Create a plot for the accuracy if recorded
if accuracy is not None and val_accuracy is not None:
    plt.subplot(1, 2, 2)
    plt.plot(epochs, accuracy, 'r-', label='Training Accuracy',
linewidth=2)
    plt.plot(epochs, val_accuracy, 'm--', label='Validation Accuracy',
linewidth=2)
    plt.title('Training and Validation Accuracy using CNN')
    plt.xlabel('Epochs')
    plt.ylabel('Accuracy')
    plt.legend()
    plt.grid(True)

plt.tight_layout()
plt.show()
```



```
import matplotlib.pyplot as plt

# Define the epoch data
epochs = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
train_loss = [864.8176, 751.2241, 236.9434, 100.0475, 36.8922,
16.9396, 3.5025, 2.2461, 2.2943, 2.2931]
train_accuracy = [0.1857, 0.2143, 0.2071, 0.3429, 0.3929, 0.4286,
0.3071, 0.1643, 0.1286, 0.1286]
val_loss = [732.0569, 184.1177, 45.3959, 71.9916, 39.2038, 4.3505,
2.2363, 2.3018, 2.3015, 2.3012]
val_accuracy = [0.3143, 0.2571, 0.3429, 0.2857, 0.3714, 0.2571,
0.1143, 0.0857, 0.0857, 0.0857]

# Plot training and validation loss
plt.figure(figsize=(10, 5))
plt.plot(epochs, train_loss, 'b-o', label='Training loss')
plt.plot(epochs, val_loss, 'r-o', label='Validation loss')
plt.title('Training and Validation Loss using CNN')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.grid(True)
plt.show()

# Plot training and validation accuracy
plt.figure(figsize=(10, 5))
plt.plot(epochs, train_accuracy, 'b-o', label='Training accuracy')
plt.plot(epochs, val_accuracy, 'r-o', label='Validation accuracy')
plt.title('Training and Validation Accuracy using CNN')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.grid(True)
plt.show()
```





Xception

```
base_model = tf.keras.applications.Xception(weights='imagenet',
include_top=False, input_shape=(224, 224, 3))
x = base_model.output
x = tf.keras.layers.GlobalAveragePooling2D()(x)
x = tf.keras.layers.Dense(10, activation='softmax')(x)
xception = tf.keras.Model(inputs=base_model.input, outputs=x)


xception.compile(optimizer=tf.keras.optimizers.Adam(0.0001),

loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

Xception = xception.fit(train_ds, validation_data=val_ds, epochs=15,
    callbacks = [
        tf.keras.callbacks.EarlyStopping(
            monitor="val_loss",
            min_delta=1e-2,
            patience=3,
            verbose=1,
        )
    ]
)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-
applications/xception/
xception_weights_tf_dim_ordering_tf_kernels_notop.h5
83683744/83683744 [==============================] - 1s 0us/step
Epoch 1/15

/usr/local/lib/python3.10/dist-packages/keras/src/backend.py:5727:
UserWarning: "`sparse_categorical_crossentropy` received
`from_logits=True`, but the `output` argument was produced by a
Softmax activation and thus does not represent logits. Was this
intended?
  output, from_logits = _get_logits(

5/5 [==============================] - 159s 27s/step - loss: 2.3072 -
accuracy: 0.1643 - val_loss: 2.4716 - val_accuracy: 0.0857
Epoch 2/15
5/5 [==============================] - 132s 25s/step - loss: 1.5736 -
accuracy: 0.8571 - val_loss: 2.2588 - val_accuracy: 0.2286
Epoch 3/15
5/5 [==============================] - 131s 25s/step - loss: 1.0191 -
accuracy: 0.9714 - val_loss: 2.1389 - val_accuracy: 0.2857
Epoch 4/15
5/5 [==============================] - 130s 25s/step - loss: 0.6496 -
accuracy: 0.9786 - val_loss: 2.1333 - val_accuracy: 0.3143
Epoch 5/15
5/5 [==============================] - 132s 25s/step - loss: 0.4299 -
```

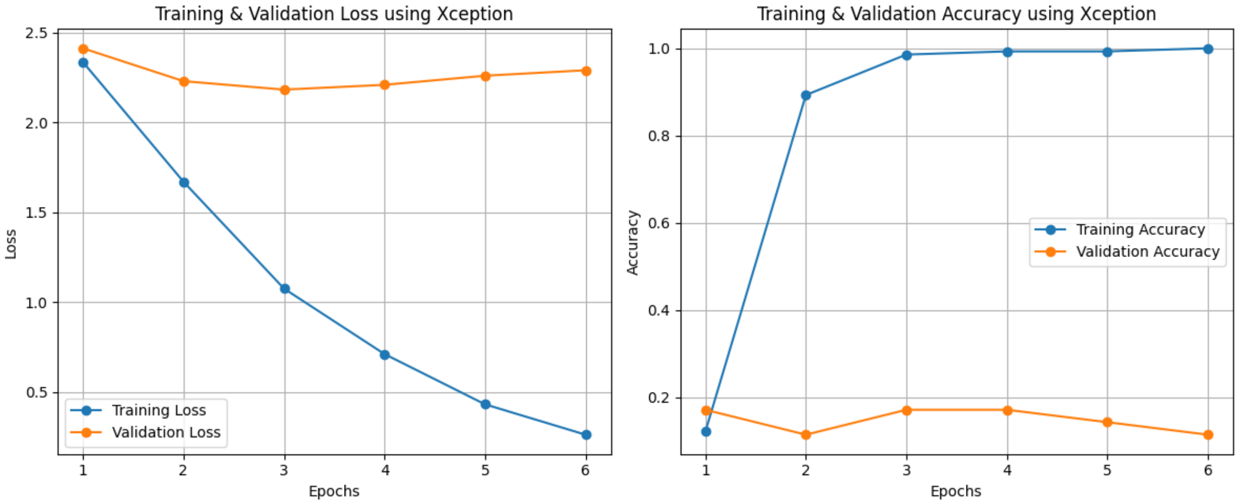
```
accuracy: 0.9857 - val_loss: 2.1595 - val_accuracy: 0.2857
Epoch 6/15
5/5 [==============================] - 140s 27s/step - loss: 0.3155 -
accuracy: 0.9857 - val_loss: 2.1224 - val_accuracy: 0.3143
Epoch 7/15
5/5 [==============================] - 132s 26s/step - loss: 0.1802 -
accuracy: 1.0000 - val_loss: 2.0666 - val_accuracy: 0.3429
Epoch 8/15
5/5 [==============================] - 131s 26s/step - loss: 0.1191 -
accuracy: 1.0000 - val_loss: 1.9788 - val_accuracy: 0.4000
Epoch 9/15
5/5 [==============================] - 131s 26s/step - loss: 0.0821 -
accuracy: 1.0000 - val_loss: 1.9116 - val_accuracy: 0.4000
Epoch 10/15
5/5 [==============================] - 134s 26s/step - loss: 0.0694 -
accuracy: 1.0000 - val_loss: 1.8287 - val_accuracy: 0.4000
Epoch 11/15
5/5 [==============================] - 132s 25s/step - loss: 0.0452 -
accuracy: 1.0000 - val_loss: 1.7711 - val_accuracy: 0.4286
Epoch 12/15
5/5 [==============================] - 128s 25s/step - loss: 0.0338 -
accuracy: 1.0000 - val_loss: 1.7085 - val_accuracy: 0.4286
Epoch 13/15
5/5 [==============================] - 129s 25s/step - loss: 0.0289 -
accuracy: 1.0000 - val_loss: 1.6646 - val_accuracy: 0.4286
Epoch 14/15
5/5 [==============================] - 127s 25s/step - loss: 0.0232 -
accuracy: 1.0000 - val_loss: 1.6180 - val_accuracy: 0.4286
Epoch 15/15
5/5 [==============================] - 137s 27s/step - loss: 0.0192 -
accuracy: 1.0000 - val_loss: 1.5722 - val_accuracy: 0.4571
```

```
import numpy as np

predictions = xception.predict(train_ds)
damage_masks = predictions > 0.5
damage_percentages = [np.mean(mask) * 100 for mask in damage_masks]
average_damage_percentage = np.mean(damage_percentages)
print(f"Average Damage Percentage: {average_damage_percentage:.2f}%")
```

```
5/5 [==============================] - 31s 5s/step
Average Damage Percentage: 9.14%
```

```
import matplotlib.pyplot as plt

# Data for epochs, training and validation loss, and accuracy
epochs = [1, 2, 3, 4, 5, 6]
train_loss = [2.3346, 1.6699, 1.0760, 0.7120, 0.4327, 0.2639]
train_accuracy = [0.1214, 0.8929, 0.9857, 0.9929, 0.9929, 1.0000]
val_loss = [2.4116, 2.2291, 2.1820, 2.2086, 2.2592, 2.2895]
```

```
val_accuracy = [0.1714, 0.1143, 0.1714, 0.1714, 0.1429, 0.1143]

# Create a figure with subplots for loss and accuracy
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 5))

# Plotting training and validation loss on the first subplot
ax1.plot(epochs, train_loss, label='Training Loss', marker='o')
ax1.plot(epochs, val_loss, label='Validation Loss', marker='o')
ax1.set_title('Training & Validation Loss using Xception')
ax1.set_xlabel('Epochs')
ax1.set_ylabel('Loss')
ax1.legend()
ax1.grid(True)

# Plotting training and validation accuracy on the second subplot
ax2.plot(epochs, train_accuracy, label='Training Accuracy',
marker='o')
ax2.plot(epochs, val_accuracy, label='Validation Accuracy',
marker='o')
ax2.set_title('Training & Validation Accuracy using Xception')
ax2.set_xlabel('Epochs')
ax2.set_ylabel('Accuracy')
ax2.legend()
ax2.grid(True)

# Show the plot
plt.tight_layout()
plt.show()
```



DenseNet121

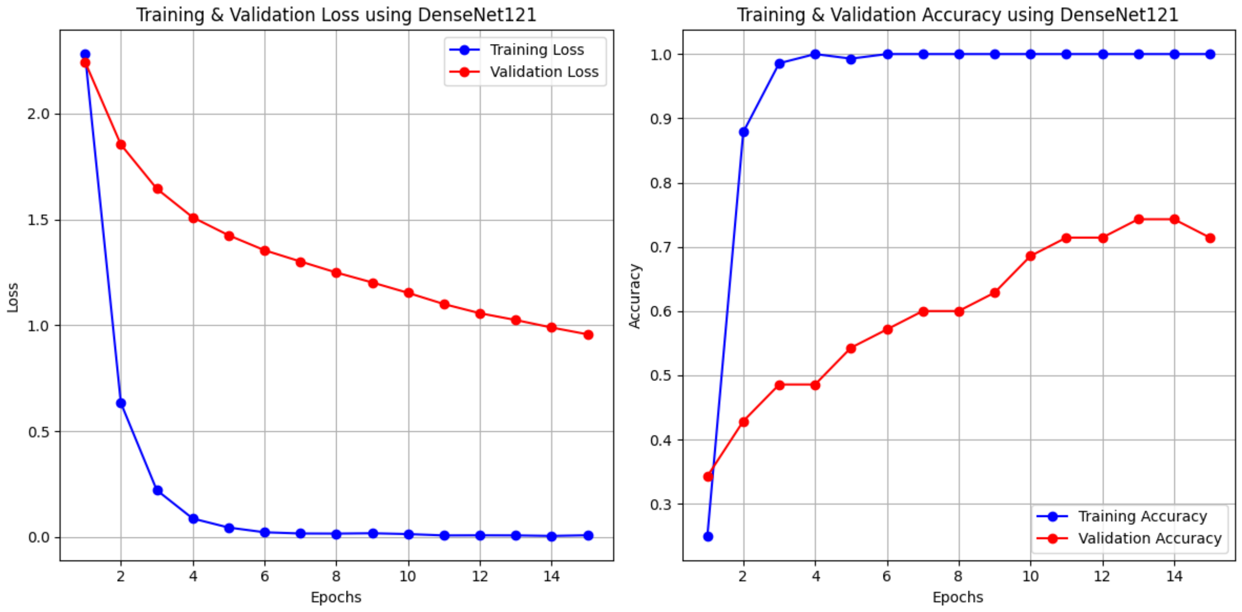
```
DenseNet121 = tf.keras.applications.DenseNet121(weights='imagenet',
include_top=False, input_shape=(224, 224, 3))
x = DenseNet121.output
```

```
x = tf.keras.layers.GlobalAveragePooling2D()(x)
x = tf.keras.layers.Dense(10, activation='softmax')(x)
new1 = tf.keras.Model(inputs=DenseNet121.input, outputs=x)

new1.compile(optimizer=tf.keras.optimizers.Adam(0.0001),

loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

DenseNet121 = new1.fit(train_ds, validation_data=val_ds, epochs=15,
    callbacks = [
        tf.keras.callbacks.EarlyStopping(
            monitor="val_loss",
            min_delta=1e-2,
            patience=3,
            verbose=1,
        )
    ]
)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-
applications/densenet/
densenet121_weights_tf_dim_ordering_tf_kernels_notop.h5
29084464/29084464 [==============================] - 0s 0us/step
Epoch 1/15

/usr/local/lib/python3.10/dist-packages/keras/src/backend.py:5727:
UserWarning: "`sparse_categorical_crossentropy` received
`from_logits=True`, but the `output` argument was produced by a
Softmax activation and thus does not represent logits. Was this
intended?
  output, from_logits = _get_logits(

5/5 [==============================] - 158s 22s/step - loss: 2.1499 -
accuracy: 0.2071 - val_loss: 2.0875 - val_accuracy: 0.3429
Epoch 2/15
5/5 [==============================] - 110s 21s/step - loss: 0.5345 -
accuracy: 0.9357 - val_loss: 1.7505 - val_accuracy: 0.4571
Epoch 3/15
5/5 [==============================] - 110s 21s/step - loss: 0.1563 -
accuracy: 1.0000 - val_loss: 1.6178 - val_accuracy: 0.5143
Epoch 4/15
5/5 [==============================] - 113s 22s/step - loss: 0.0769 -
accuracy: 1.0000 - val_loss: 1.5545 - val_accuracy: 0.4857
Epoch 5/15
5/5 [==============================] - 105s 20s/step - loss: 0.0584 -
accuracy: 1.0000 - val_loss: 1.4802 - val_accuracy: 0.5143
Epoch 6/15
5/5 [==============================] - 110s 21s/step - loss: 0.0308 -
accuracy: 1.0000 - val_loss: 1.4140 - val_accuracy: 0.5429
```

```
Epoch 7/15
5/5 [==============================] - 104s 20s/step - loss: 0.0152 -
accuracy: 1.0000 - val_loss: 1.3544 - val_accuracy: 0.5143
Epoch 8/15
5/5 [==============================] - 106s 20s/step - loss: 0.0133 -
accuracy: 1.0000 - val_loss: 1.2989 - val_accuracy: 0.5714
Epoch 9/15
5/5 [==============================] - 109s 21s/step - loss: 0.0138 -
accuracy: 1.0000 - val_loss: 1.2350 - val_accuracy: 0.6000
Epoch 10/15
5/5 [==============================] - 111s 22s/step - loss: 0.0091 -
accuracy: 1.0000 - val_loss: 1.1815 - val_accuracy: 0.6000
Epoch 11/15
5/5 [==============================] - 108s 21s/step - loss: 0.0115 -
accuracy: 1.0000 - val_loss: 1.1399 - val_accuracy: 0.6286
Epoch 12/15
5/5 [==============================] - 105s 20s/step - loss: 0.0095 -
accuracy: 1.0000 - val_loss: 1.1053 - val_accuracy: 0.6857
Epoch 13/15
5/5 [==============================] - 106s 21s/step - loss: 0.0067 -
accuracy: 1.0000 - val_loss: 1.0702 - val_accuracy: 0.6857
Epoch 14/15
5/5 [==============================] - 107s 21s/step - loss: 0.0054 -
accuracy: 1.0000 - val_loss: 1.0388 - val_accuracy: 0.6571
Epoch 15/15
5/5 [==============================] - 115s 22s/step - loss: 0.0065 -
accuracy: 1.0000 - val_loss: 1.0079 - val_accuracy: 0.6571

import numpy as np

predictions = new1.predict(train_ds)
damage_masks = predictions > 0.5
damage_percentages = [np.mean(mask) * 100 for mask in damage_masks]
average_damage_percentage = np.mean(damage_percentages)
print(f"Average Damage Percentage: {average_damage_percentage:.2f}%")

5/5 [==============================] - 24s 4s/step
Average Damage Percentage: 9.93%

import matplotlib.pyplot as plt

# Define the epoch data
epochs = list(range(1, 16))  # 1 to 15
train_loss = [2.2811, 0.6329, 0.2212, 0.0874, 0.0447, 0.0223, 0.0166,
0.0161, 0.0177, 0.0138, 0.0074, 0.0080, 0.0076, 0.0051, 0.0083]
train_accuracy = [0.2500, 0.8786, 0.9857, 1.0000, 0.9929, 1.0000,
1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000,
1.0000]
val_loss = [2.2447, 1.8535, 1.6451, 1.5100, 1.4252, 1.3546, 1.3015,
1.2493, 1.2024, 1.1532, 1.1002, 1.0570, 1.0249, 0.9894, 0.9574]
```

```python
val_accuracy = [0.3429, 0.4286, 0.4857, 0.4857, 0.5429, 0.5714,
0.6000, 0.6000, 0.6286, 0.6857, 0.7143, 0.7143, 0.7429, 0.7429,
0.7143]

# Plot training and validation loss
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.plot(epochs, train_loss, 'bo-', label='Training Loss')
plt.plot(epochs, val_loss, 'ro-', label='Validation Loss')
plt.title('Training & Validation Loss using DenseNet121')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.grid(True)

# Plot training and validation accuracy
plt.subplot(1, 2, 2)
plt.plot(epochs, train_accuracy, 'bo-', label='Training Accuracy')
plt.plot(epochs, val_accuracy, 'ro-', label='Validation Accuracy')
plt.title('Training & Validation Accuracy using DenseNet121')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.grid(True)

# Show the plots
plt.tight_layout()
plt.show()
```



VGG16

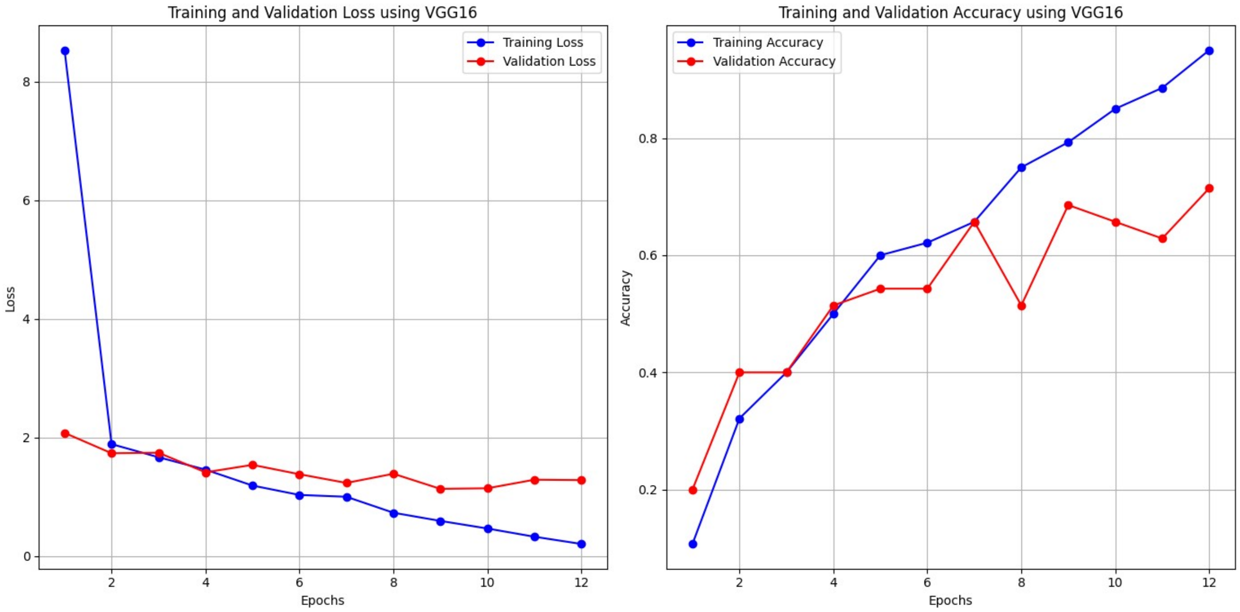
```
base_model = tf.keras.applications.VGG16(weights='imagenet',
include_top=False, input_shape=(224, 224, 3))
x = base_model.output
x = tf.keras.layers.Flatten()(x)
x = tf.keras.layers.Dense(128, activation='relu')(x)
x = tf.keras.layers.Dropout(0.2)(x)
x = tf.keras.layers.Dense(10, activation='softmax')(x)
VGG16 = tf.keras.Model(inputs=base_model.input, outputs=x)

VGG16.compile(optimizer=tf.keras.optimizers.Adam(0.0001),

loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

VGG16N = VGG16.fit(train_ds, validation_data=val_ds, epochs=15,
    callbacks = [
        tf.keras.callbacks.EarlyStopping(
            monitor="val_loss",
            min_delta=1e-2,
            patience=3,
            verbose=1,
        )
    ]
)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-
applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58889256/58889256 [==============================] - 0s 0us/step
Epoch 1/15
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/backend.py:5727:
UserWarning: "`sparse_categorical_crossentropy` received
`from_logits=True`, but the `output` argument was produced by a
Softmax activation and thus does not represent logits. Was this
intended?
  output, from_logits = _get_logits(
```

```
5/5 [==============================] - 290s 56s/step - loss: 10.1455 -
accuracy: 0.1143 - val_loss: 1.9568 - val_accuracy: 0.2000
Epoch 2/15
5/5 [==============================] - 282s 55s/step - loss: 1.9626 -
accuracy: 0.3286 - val_loss: 1.8059 - val_accuracy: 0.4571
Epoch 3/15
5/5 [==============================] - 287s 56s/step - loss: 1.6420 -
accuracy: 0.4143 - val_loss: 1.6413 - val_accuracy: 0.4286
Epoch 4/15
5/5 [==============================] - 287s 57s/step - loss: 1.1600 -
accuracy: 0.5714 - val_loss: 1.2836 - val_accuracy: 0.4857
Epoch 5/15
5/5 [==============================] - 289s 57s/step - loss: 0.9268 -
```

```
accuracy: 0.6500 - val_loss: 1.5949 - val_accuracy: 0.4857
Epoch 6/15
5/5 [==============================] - 309s 61s/step - loss: 0.9446 -
accuracy: 0.6643 - val_loss: 1.0147 - val_accuracy: 0.6857
Epoch 7/15
5/5 [==============================] - 286s 56s/step - loss: 0.6591 -
accuracy: 0.7643 - val_loss: 0.9348 - val_accuracy: 0.6857
Epoch 8/15
5/5 [==============================] - 290s 57s/step - loss: 0.4260 -
accuracy: 0.8500 - val_loss: 1.1876 - val_accuracy: 0.6000
Epoch 9/15
5/5 [==============================] - 288s 56s/step - loss: 0.3742 -
accuracy: 0.8714 - val_loss: 1.0324 - val_accuracy: 0.7143
Epoch 10/15
5/5 [==============================] - 288s 56s/step - loss: 0.2315 -
accuracy: 0.9286 - val_loss: 1.0973 - val_accuracy: 0.6857
Epoch 10: early stopping

import numpy as np

predictions = VGG16.predict(train_ds)
damage_masks = predictions > 0.5
damage_percentages = [np.mean(mask) * 100 for mask in damage_masks]
average_damage_percentage = np.mean(damage_percentages)
print(f"Average Damage Percentage: {average_damage_percentage:.2f}%")

5/5 [==============================] - 135s 14s/step
Average Damage Percentage: 9.57%

import matplotlib.pyplot as plt

# Epoch data
epochs = range(1, 13)  # Since training stopped at epoch 12
train_loss = [8.5268, 1.8877, 1.6663, 1.4570, 1.1889, 1.0306, 0.9999,
0.7309, 0.5929, 0.4651, 0.3266, 0.2047]
train_accuracy = [0.1071, 0.3214, 0.4000, 0.5000, 0.6000, 0.6214,
0.6571, 0.7500, 0.7929, 0.8500, 0.8857, 0.9500]
val_loss = [2.0763, 1.7338, 1.7421, 1.4123, 1.5376, 1.3797, 1.2331,
1.3877, 1.1332, 1.1426, 1.2873, 1.2809]
val_accuracy = [0.2000, 0.4000, 0.4000, 0.5143, 0.5429, 0.5429,
0.6571, 0.5143, 0.6857, 0.6571, 0.6286, 0.7143]

# Creating the plots
plt.figure(figsize=(14, 7))

# Plot training and validation loss
plt.subplot(1, 2, 1)
plt.plot(epochs, train_loss, 'b-o', label='Training Loss')
plt.plot(epochs, val_loss, 'r-o', label='Validation Loss')
plt.title('Training and Validation Loss using VGG16')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.grid(True)

# Plot training and validation accuracy
plt.subplot(1, 2, 2)
plt.plot(epochs, train_accuracy, 'b-o', label='Training Accuracy')
plt.plot(epochs, val_accuracy, 'r-o', label='Validation Accuracy')
plt.title('Training and Validation Accuracy using VGG16')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.grid(True)

# Show the plots
plt.tight_layout()
plt.show()
```

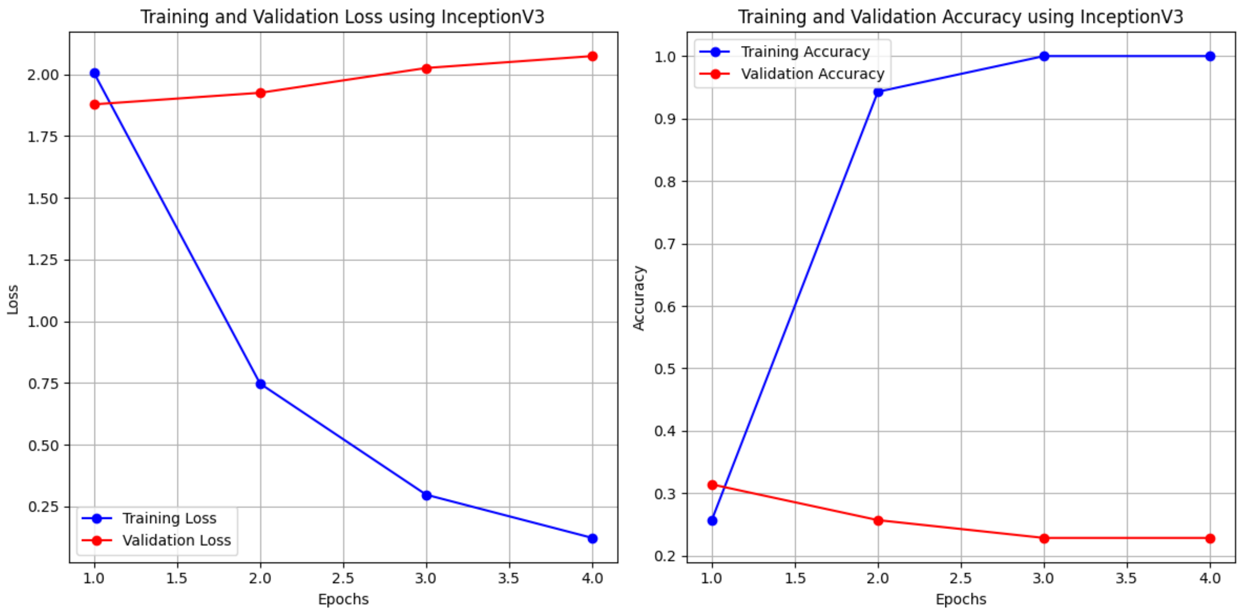

Training and Validation Loss using VGG16

Training and Validation Accuracy using VGG16

InceptionV3

```
base_model = tf.keras.applications.InceptionV3(weights='imagenet',
include_top=False, input_shape=(224, 224, 3))
x = base_model.output
x = tf.keras.layers.GlobalAveragePooling2D()(x)
x = tf.keras.layers.Dense(10, activation='softmax')(x)
model = tf.keras.Model(inputs=base_model.input, outputs=x)

model.compile(optimizer=tf.keras.optimizers.Adam(0.0001),
loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

InceptionV3 = model.fit(train_ds, validation_data=val_ds, epochs=15,
    callbacks = [
        tf.keras.callbacks.EarlyStopping(
            monitor="val_loss",
            min_delta=1e-2,
            patience=3,
            verbose=1,
        )
    ]
)

Downloading data from https://storage.googleapis.com/tensorflow/keras-
applications/inception_v3/
inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5
87910968/87910968 [==============================] - 5s 0us/step
Epoch 1/15
5/5 [==============================] - 98s 14s/step - loss: 2.1240 -
accuracy: 0.2143 - val_loss: 2.1485 - val_accuracy: 0.2571
Epoch 2/15
5/5 [==============================] - 67s 13s/step - loss: 0.8094 -
accuracy: 0.9571 - val_loss: 1.9144 - val_accuracy: 0.2857
Epoch 3/15
5/5 [==============================] - 68s 13s/step - loss: 0.3040 -
accuracy: 0.9857 - val_loss: 1.9089 - val_accuracy: 0.2571
Epoch 4/15
5/5 [==============================] - 71s 14s/step - loss: 0.1060 -
accuracy: 1.0000 - val_loss: 1.9418 - val_accuracy: 0.3143
Epoch 5/15
5/5 [==============================] - 68s 13s/step - loss: 0.0459 -
accuracy: 1.0000 - val_loss: 1.9413 - val_accuracy: 0.3143
Epoch 5: early stopping

import numpy as np

predictions = model.predict(train_ds)
damage_masks = predictions > 0.5
damage_percentages = [np.mean(mask) * 100 for mask in damage_masks]
average_damage_percentage = np.mean(damage_percentages)
print(f"Average Damage Percentage: {average_damage_percentage:.2f}%")

5/5 [==============================] - 68s 3s/step
Average Damage Percentage: 4.71%

import matplotlib.pyplot as plt

# Define the data for epochs, training and validation loss, and
accuracy
```

```
epochs = [1, 2, 3, 4]
train_loss = [2.0063, 0.7475, 0.2974, 0.1224]
train_accuracy = [0.2571, 0.9429, 1.0000, 1.0000]
val_loss = [1.8789, 1.9256, 2.0261, 2.0744]
val_accuracy = [0.3143, 0.2571, 0.2286, 0.2286]

# Create a figure with subplots for loss and accuracy
plt.figure(figsize=(12, 6))

# Plot training and validation loss
plt.subplot(1, 2, 1)
plt.plot(epochs, train_loss, 'b-o', label='Training Loss')
plt.plot(epochs, val_loss, 'r-o', label='Validation Loss')
plt.title('Training and Validation Loss using InceptionV3')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.grid(True)

# Plot training and validation accuracy
plt.subplot(1, 2, 2)
plt.plot(epochs, train_accuracy, 'b-o', label='Training Accuracy')
plt.plot(epochs, val_accuracy, 'r-o', label='Validation Accuracy')
plt.title('Training and Validation Accuracy using InceptionV3')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.grid(True)

# Show the plot
plt.tight_layout()
plt.show()
```



EfficientNetB0

```
EfficientNetB0 =
tf.keras.applications.EfficientNetB0(weights='imagenet',
include_top=False, input_shape=(224, 224, 3))
x = EfficientNetB0.output
x = tf.keras.layers.GlobalAveragePooling2D()(x)
x = tf.keras.layers.Dense(10, activation='softmax')(x)
efficientNetB0 = tf.keras.Model(inputs=EfficientNetB0.input,
outputs=x)

efficientNetB0.compile(optimizer=tf.keras.optimizers.Adam(0.0001),

loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

EfficientNetB0 = efficientNetB0.fit(train_ds, validation_data=val_ds,
epochs=15,
    callbacks = [
        tf.keras.callbacks.EarlyStopping(
            monitor="val_loss",
            min_delta=1e-2,
            patience=3,
            verbose=1,
        )
    ]
)
```

```
Downloading data from https://storage.googleapis.com/keras-
applications/efficientnetb0_notop.h5
```

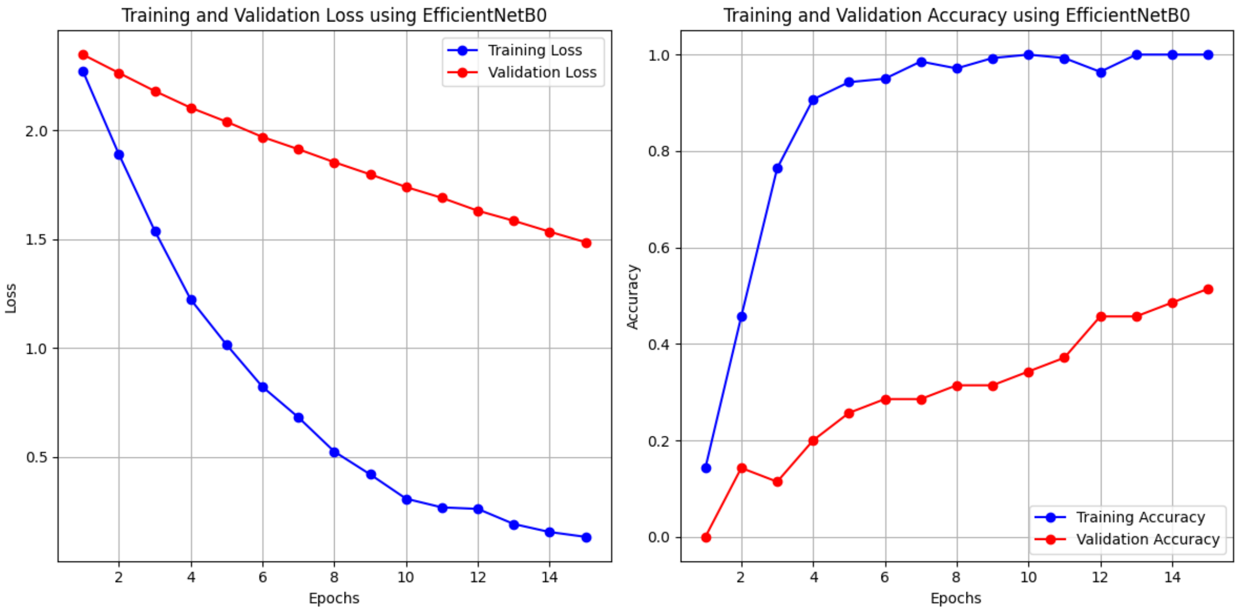
```
16705208/16705208 [==============================] - 0s 0us/step
Epoch 1/15

/usr/local/lib/python3.10/dist-packages/keras/src/backend.py:5727:
UserWarning: "`sparse_categorical_crossentropy` received
`from_logits=True`, but the `output` argument was produced by a
Softmax activation and thus does not represent logits. Was this
intended?
  output, from_logits = _get_logits(

5/5 [==============================] - 77s 9s/step - loss: 2.2992 -
accuracy: 0.1429 - val_loss: 2.4348 - val_accuracy: 0.0286
Epoch 2/15
5/5 [==============================] - 47s 9s/step - loss: 1.9062 -
accuracy: 0.4071 - val_loss: 2.3544 - val_accuracy: 0.0857
Epoch 3/15
5/5 [==============================] - 46s 9s/step - loss: 1.5627 -
accuracy: 0.7286 - val_loss: 2.2749 - val_accuracy: 0.1429
Epoch 4/15
5/5 [==============================] - 45s 9s/step - loss: 1.2987 -
accuracy: 0.8857 - val_loss: 2.1927 - val_accuracy: 0.2000
Epoch 5/15
5/5 [==============================] - 48s 9s/step - loss: 1.1087 -
accuracy: 0.8786 - val_loss: 2.1077 - val_accuracy: 0.2286
Epoch 6/15
5/5 [==============================] - 43s 8s/step - loss: 0.9103 -
accuracy: 0.9571 - val_loss: 2.0307 - val_accuracy: 0.3143
Epoch 7/15
5/5 [==============================] - 44s 8s/step - loss: 0.7437 -
accuracy: 0.9857 - val_loss: 1.9663 - val_accuracy: 0.3143
Epoch 8/15
5/5 [==============================] - 46s 9s/step - loss: 0.6005 -
accuracy: 0.9786 - val_loss: 1.9000 - val_accuracy: 0.3143
Epoch 9/15
5/5 [==============================] - 47s 9s/step - loss: 0.5321 -
accuracy: 0.9857 - val_loss: 1.8299 - val_accuracy: 0.3429
Epoch 10/15
5/5 [==============================] - 51s 10s/step - loss: 0.3743 -
accuracy: 1.0000 - val_loss: 1.7722 - val_accuracy: 0.3714
Epoch 11/15
5/5 [==============================] - 45s 8s/step - loss: 0.3607 -
accuracy: 1.0000 - val_loss: 1.7126 - val_accuracy: 0.4000
Epoch 12/15
5/5 [==============================] - 48s 9s/step - loss: 0.3021 -
accuracy: 1.0000 - val_loss: 1.6527 - val_accuracy: 0.4000
Epoch 13/15
5/5 [==============================] - 48s 9s/step - loss: 0.2462 -
accuracy: 1.0000 - val_loss: 1.5902 - val_accuracy: 0.4000
Epoch 14/15
5/5 [==============================] - 47s 9s/step - loss: 0.2084 -
```

```
accuracy: 1.0000 - val_loss: 1.5284 - val_accuracy: 0.4286
Epoch 15/15
5/5 [==============================] - 48s 9s/step - loss: 0.1704 -
accuracy: 0.9929 - val_loss: 1.4634 - val_accuracy: 0.4571
```

```
import numpy as np

predictions = efficientNetB0.predict(train_ds)
damage_masks = predictions > 0.5
damage_percentages = [np.mean(mask) * 100 for mask in damage_masks]
average_damage_percentage = np.mean(damage_percentages)
print(f"Average Damage Percentage: {average_damage_percentage:.2f}%")
```

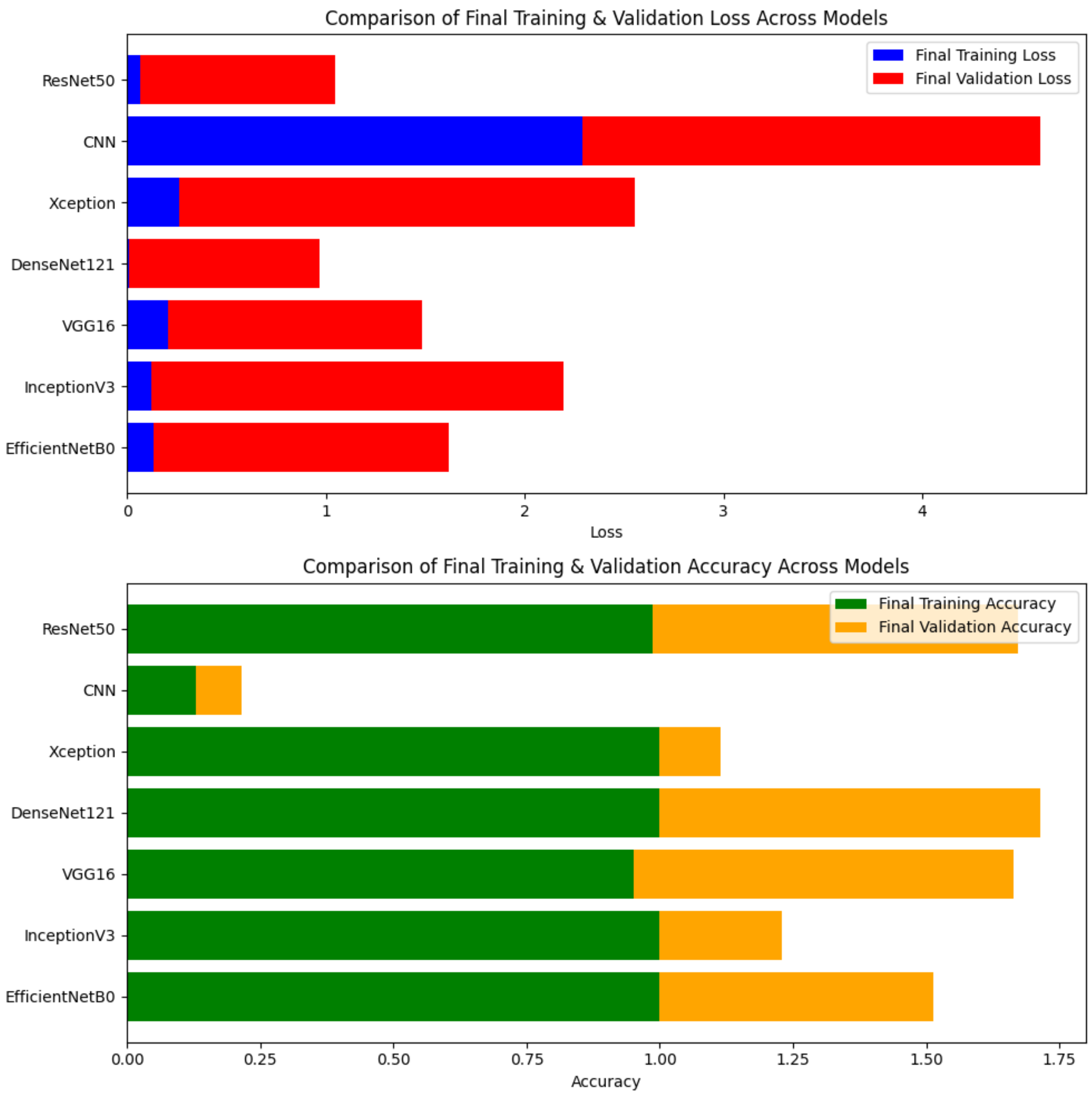
```
5/5 [==============================] - 13s 2s/step
Average Damage Percentage: 7.07%
```

```
import matplotlib.pyplot as plt

# Data for epochs, training and validation loss, and accuracy
epochs = range(1, 16)  # 1 to 15
train_loss = [2.2712, 1.8907, 1.5387, 1.2241, 1.0161, 0.8228, 0.6828,
0.5256, 0.4208, 0.3091, 0.2692, 0.2623, 0.1930, 0.1561, 0.1339]
train_accuracy = [0.1429, 0.4571, 0.7643, 0.9071, 0.9429, 0.9500,
0.9857, 0.9714, 0.9929, 1.0000, 0.9929, 0.9643, 1.0000, 1.0000,
1.0000]
val_loss = [2.3482, 2.2640, 2.1809, 2.1044, 2.0399, 1.9700, 1.9138,
1.8543, 1.7984, 1.7404, 1.6904, 1.6313, 1.5853, 1.5352, 1.4867]
val_accuracy = [0.0000, 0.1429, 0.1143, 0.2000, 0.2571, 0.2857,
0.2857, 0.3143, 0.3143, 0.3429, 0.3714, 0.4571, 0.4571, 0.4857,
0.5143]

# Create a figure with subplots for loss and accuracy
plt.figure(figsize=(12, 6))

# Plot training and validation loss
plt.subplot(1, 2, 1)
plt.plot(epochs, train_loss, 'b-o', label='Training Loss')
plt.plot(epochs, val_loss, 'r-o', label='Validation Loss')
plt.title('Training and Validation Loss using EfficientNetB0')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.grid(True)

# Plot training and validation accuracy
plt.subplot(1, 2, 2)
plt.plot(epochs, train_accuracy, 'b-o', label='Training Accuracy')
plt.plot(epochs, val_accuracy, 'r-o', label='Validation Accuracy')
plt.title('Training and Validation Accuracy using EfficientNetB0')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.grid(True)

# Show the plots
plt.tight_layout()
plt.show()
```



```
import matplotlib.pyplot as plt

# Define the data from final epochs for each model
models = ['ResNet50', 'CNN', 'Xception', 'DenseNet121', 'VGG16',
'InceptionV3', 'EfficientNetB0']
final_train_loss = [0.0651, 2.2931, 0.2639, 0.0083, 0.2047, 0.1224,
0.1339]
final_val_loss = [0.9830, 2.3012, 2.2895, 0.9574, 1.2809, 2.0744,
1.4867]
final_train_accuracy = [0.9857, 0.1286, 1.0000, 1.0000, 0.9500,
1.0000, 1.0000]
final_val_accuracy = [0.6857, 0.0857, 0.1143, 0.7143, 0.7143, 0.2286,
0.5143]

fig, ax = plt.subplots(2, 1, figsize=(10, 10))

# Plot for final training and validation loss
ax[0].barh(models, final_train_loss, color='blue', label='Final
Training Loss')
ax[0].barh(models, final_val_loss, color='red', left=final_train_loss,
label='Final Validation Loss')
ax[0].set_xlabel('Loss')
```

```python
ax[0].set_title('Comparison of Final Training & Validation Loss Across Models')
ax[0].invert_yaxis()  # Invert y-axis to have the first model at the top
ax[0].legend()

# Plot for final training and validation accuracy
ax[1].barh(models, final_train_accuracy, color='green', label='Final Training Accuracy')
ax[1].barh(models, final_val_accuracy, color='orange', left=final_train_accuracy, label='Final Validation Accuracy')
ax[1].set_xlabel('Accuracy')
ax[1].set_title('Comparison of Final Training & Validation Accuracy Across Models')
ax[1].invert_yaxis()  # Keep the y-axis inverted
ax[1].legend()

plt.tight_layout()
plt.show()
```



```python
import pandas as pd

# Create a DataFrame from the data
data = {
    "Model": models,
    "Final Training Loss": final_train_loss,
    "Final Validation Loss": final_val_loss,
    "Final Training Accuracy": final_train_accuracy,
    "Final Validation Accuracy": final_val_accuracy
}

df = pd.DataFrame(data)
```

```
# Display the DataFrame
print(df)

# Optionally, save the DataFrame to a CSV file
# df.to_csv("model_comparison.csv", index=False)
```

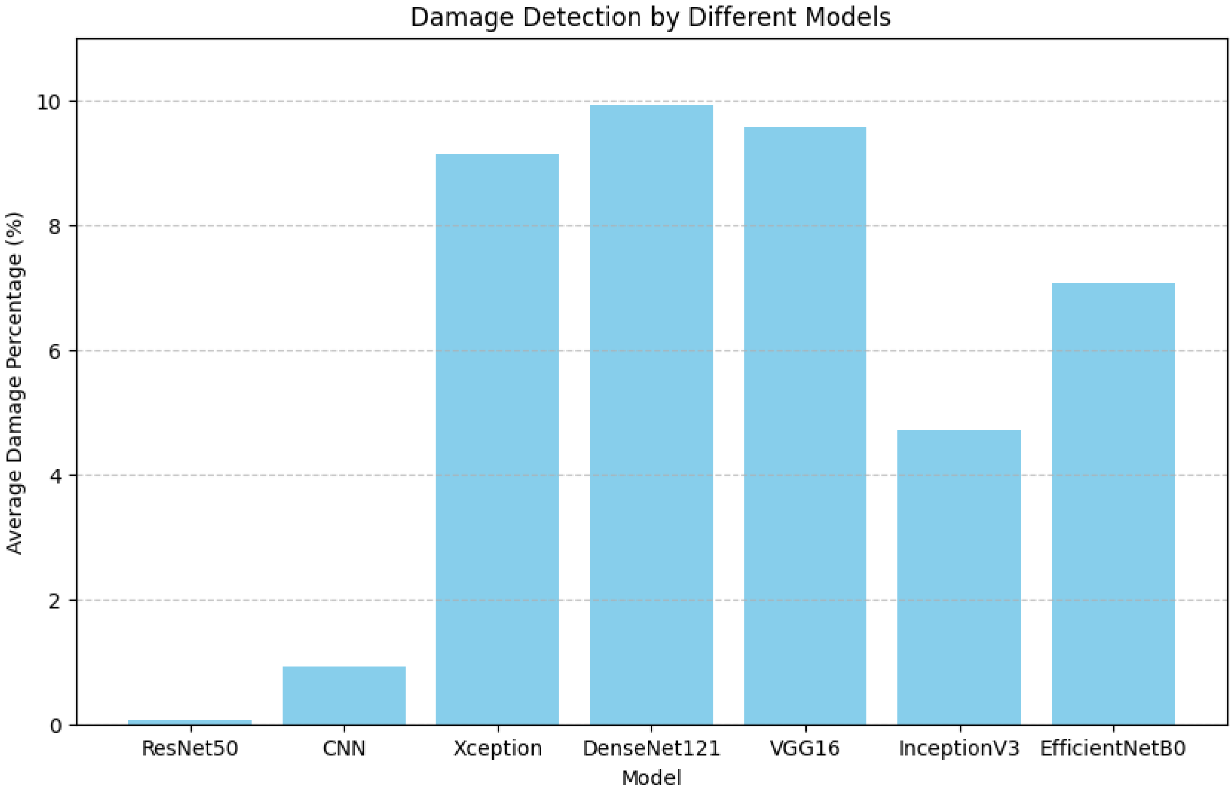
```
            Model  Final Training Loss  Final Validation Loss  \
0        ResNet50               0.0651                 0.9830
1             CNN               2.2931                 2.3012
2        Xception               0.2639                 2.2895
3     DenseNet121               0.0083                 0.9574
4           VGG16               0.2047                 1.2809
5     InceptionV3               0.1224                 2.0744
6  EfficientNetB0               0.1339                 1.4867

   Final Training Accuracy  Final Validation Accuracy
0                   0.9857                     0.6857
1                   0.1286                     0.0857
2                   1.0000                     0.1143
3                   1.0000                     0.7143
4                   0.9500                     0.7143
5                   1.0000                     0.2286
6                   1.0000                     0.5143
```

```
pip install matplotlib pandas
```

```
Requirement already satisfied: matplotlib in
/usr/local/lib/python3.10/dist-packages (3.7.1)
Requirement already satisfied: pandas in
/usr/local/lib/python3.10/dist-packages (2.0.3)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.2.1)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (4.51.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.5)
Requirement already satisfied: numpy>=1.20 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.25.2)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (24.0)
Requirement already satisfied: pillow>=6.2.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
```

```
/usr/local/lib/python3.10/dist-packages (from pandas) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2024.1)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7-
>matplotlib) (1.16.0)
```

```python
import matplotlib.pyplot as plt
import pandas as pd

# Define the data
data = {
    "Model": ["ResNet50", "CNN", "Xception", "DenseNet121", "VGG16",
"InceptionV3", "EfficientNetB0"],
    "Average Damage Percentage": [0.07, 0.93, 9.14, 9.93, 9.57, 4.71,
7.07]
}

# Create a DataFrame
df = pd.DataFrame(data)

# Plotting
plt.figure(figsize=(10, 6))  # Set the figure size
plt.bar(df["Model"], df["Average Damage Percentage"], color='skyblue')
plt.xlabel('Model')
plt.ylabel('Average Damage Percentage (%)')
plt.title('Damage Detection by Different Models')
plt.ylim(0, 11)  # Extend y-axis limit to better visualize differences
plt.grid(axis='y', linestyle='--', alpha=0.7)  # Add gridlines for
better readability
plt.show()
```


Damage Detection by Different Models
Average Damage Percentage (%)
0
2
4
6
8
10
ResNet50
CNN
Xception
DenseNet121
VGG16
InceptionV3
EfficientNetB0
Model