```
from google.colab import drive
drive.mount('/content/drive')
```

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

```
# Install necessary libraries
!pip install tensorflow kagglehub opencv-python matplotlib scikit-learn pandas numpy Augmentor
```

```
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (2.0.2)
Collecting Augmentor
  Downloading Augmentor-0.2.12-py2.py3-none-any.whl.metadata (1.3 kB)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (25.2.10)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (18.1.1)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.4.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from tensorflow) (24.2)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.3 in /usr/local/lib/python3.1
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (2.32.3)
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/dist-packages (from tensorflow) (75.2.0)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.17.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.0.1)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (4.13.2)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.17.2)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.71.0)
Requirement already satisfied: tensorboard<2.19,>=2.18 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (2.18.0)
Requirement already satisfied: keras>=3.5.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.8.0)
Requirement already satisfied: h5py>=3.11.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.13.0)
Requirement already satisfied: ml-dtypes<0.5.0,>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (0.4.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (0.37.1)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.11/dist-packages (from kagglehub) (6.0.2)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from kagglehub) (4.67.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (0.12.1)
```

```
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (3.6.0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.11/dist-packages (from astunparse>=1.6.0->tensorflow) (0.45.1)
Requirement already satisfied: rich in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow) (13.9.4)
Requirement already satisfied: namex in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow) (0.0.8)
Requirement already satisfied: optree in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow) (0.15.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensorflow) (
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensorflow) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensorflow) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensorflow) (2025.1
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18->tensorflow) (3.8)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18->tensorflow) (3.1.3
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.11/dist-packages (from werkzeug>=1.0.1->tensorboard<2.19,>=2.18-
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from rich->keras>=3.5.0->tensorflow) (3.0.
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from rich->keras>=3.5.0->tensorflow) (2.
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0->rich->keras>=3.5.0->tenso
Downloading Augmentor-0.2.12-py2.py3-none-any.whl (38 kB)
Installing collected packages: Augmentor
Successfully installed Augmentor-0.2.12
```

```
!pip install ultralytics
```

```
Successfully uninstalled nvidia-nvjitlink-cu12-12.5.82
```

```
    Attempting uninstall: nvidia-curand-cu12
      Found existing installation: nvidia-curand-cu12 10.3.6.82
      Uninstalling nvidia-curand-cu12-10.3.6.82:
        Successfully uninstalled nvidia-curand-cu12-10.3.6.82
    Attempting uninstall: nvidia-cufft-cu12
      Found existing installation: nvidia-cufft-cu12 11.2.3.61
      Uninstalling nvidia-cufft-cu12-11.2.3.61:
        Successfully uninstalled nvidia-cufft-cu12-11.2.3.61
    Attempting uninstall: nvidia-cuda-runtime-cu12
      Found existing installation: nvidia-cuda-runtime-cu12 12.5.82
      Uninstalling nvidia-cuda-runtime-cu12-12.5.82:
        Successfully uninstalled nvidia-cuda-runtime-cu12-12.5.82
    Attempting uninstall: nvidia-cuda-nvrtc-cu12
      Found existing installation: nvidia-cuda-nvrtc-cu12 12.5.82
      Uninstalling nvidia-cuda-nvrtc-cu12-12.5.82:
        Successfully uninstalled nvidia-cuda-nvrtc-cu12-12.5.82
    Attempting uninstall: nvidia-cuda-cupti-cu12
      Found existing installation: nvidia-cuda-cupti-cu12 12.5.82
      Uninstalling nvidia-cuda-cupti-cu12-12.5.82:
        Successfully uninstalled nvidia-cuda-cupti-cu12-12.5.82
    Attempting uninstall: nvidia-cublas-cu12
      Found existing installation: nvidia-cublas-cu12 12.5.3.2
      Uninstalling nvidia-cublas-cu12-12.5.3.2:
        Successfully uninstalled nvidia-cublas-cu12-12.5.3.2
    Attempting uninstall: nvidia-cusparse-cu12
      Found existing installation: nvidia-cusparse-cu12 12.5.1.3
      Uninstalling nvidia-cusparse-cu12-12.5.1.3:
        Successfully uninstalled nvidia-cusparse-cu12-12.5.1.3
    Attempting uninstall: nvidia-cudnn-cu12
      Found existing installation: nvidia-cudnn-cu12 9.3.0.75
      Uninstalling nvidia-cudnn-cu12-9.3.0.75:
        Successfully uninstalled nvidia-cudnn-cu12-9.3.0.75
    Attempting uninstall: nvidia-cusolver-cu12
      Found existing installation: nvidia-cusolver-cu12 11.6.3.83
      Uninstalling nvidia-cusolver-cu12-11.6.3.83:
        Successfully uninstalled nvidia-cusolver-cu12-11.6.3.83
  Successfully installed nvidia-cublas-cu12-12.4.5.8 nvidia-cuda-cupti-cu12-12.4.127 nvidia-cuda-nvrtc-cu12-12.4.127 nvidia-cuda-runtime-cu12
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
```

```
import cv2
import shutil
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.applications import ResNet50
import kagglehub
import Augmentor
import pathlib
import glob
from IPython.display import display, Javascript
from ultralytics import YOLO
```

```
Creating new Ultralytics Settings v0.0.6 file ✅
View Ultralytics Settings with 'yolo settings' or at '/root/.config/Ultralytics/settings.json'
Update Settings with 'yolo settings key=value', i.e. 'yolo settings runs_dir=path/to/dir'. For help see https://docs.ultralytics.com/quickstar
```

```
def keep_alive():
    display(Javascript('''
        function ClickConnect(){
            console.log("Working");
            document.querySelector("colab-toolbar-button#connect").click()
        }
        setInterval(ClickConnect, 60000)
    '''))
keep_alive()
```

```
print("Downloading HAM10000 dataset...")
path = kagglehub.dataset_download("kmader/skin-cancer-mnist-ham10000")
print("Path to dataset files:", path)
data_dir_train = pathlib.Path(path)
metadata_path = os.path.join(path, "HAM10000_metadata.csv")
images_path_part1 = os.path.join(path, "HAM10000_images_part_1")
images_path_part2 = os.path.join(path, "HAM10000_images_part_2")
```

```
Downloading HAM10000 dataset...
Path to dataset files: /kaggle/input/skin-cancer-mnist-ham10000
```

```
if not os.path.exists(metadata_path):
    raise FileNotFoundError("Metadata file not found.")
if not (os.path.exists(images_path_part1) and os.path.exists(images_path_part2)):
    raise FileNotFoundError("Images directory not found.")
metadata = pd.read_csv(metadata_path)


lesion_type_dict = {
    'nv': {'name': 'Melanocytic nevi', 'severity': 'Low'},
    'mel': {'name': 'Melanoma', 'severity': 'High'},
    'bkl': {'name': 'Benign keratosis-like lesions', 'severity': 'Moderate'},
    'bcc': {'name': 'Basal cell carcinoma', 'severity': 'High'},
    'akiec': {'name': 'Actinic keratoses', 'severity': 'Moderate'},
    'vasc': {'name': 'Vascular lesions', 'severity': 'Low'},
    'df': {'name': 'Dermatofibroma', 'severity': 'Low'}
}
metadata['cell_type'] = metadata['dx'].map(lambda x: lesion_type_dict[x]['name'])
metadata['severity'] = metadata['dx'].map(lambda x: lesion_type_dict[x]['severity'])
metadata['cell_type_idx'] = pd.Categorical(metadata['cell_type']).codes
class_names = list(lesion_type_dict.keys())


restructured_dir = "/content/ham10000_restructured"
os.makedirs(restructured_dir, exist_ok=True)
for split in ['train', 'val']:
    for class_name in class_names:
        os.makedirs(os.path.join(restructured_dir, split, 'images', class_name), exist_ok=True)
        os.makedirs(os.path.join(restructured_dir, split, 'labels', class_name), exist_ok=True)

train_meta, val_meta = train_test_split(metadata, test_size=0.2, random_state=123)
for _, row in train_meta.iterrows():
    image_id = row['image_id']
    class_label = row['dx']
    src_path = os.path.join(images_path_part1, f"{image_id}.jpg")
    if not os.path.exists(src_path):
        src_path = os.path.join(images_path_part2, f"{image_id}.jpg")
    dest_path = os.path.join(restructured_dir, 'train', 'images', class_label, f"{image_id}.jpg")
    if os.path.exists(src_path):
        shutil.copy(src_path, dest_path)
        with open(os.path.join(restructured_dir, 'train', 'labels', class_label, f"{image_id}.txt"), 'w') as f:
```

```
            f.write(f"{class_names.index(class_label)} 0.5 0.5 1.0 1.0\n")

for _, row in val_meta.iterrows():
    image_id = row['image_id']
    class_label = row['dx']
    src_path = os.path.join(images_path_part1, f"{image_id}.jpg")
    if not os.path.exists(src_path):
        src_path = os.path.join(images_path_part2, f"{image_id}.jpg")
    dest_path = os.path.join(restructured_dir, 'val', 'images', class_label, f"{image_id}.jpg")
    if os.path.exists(src_path):
        shutil.copy(src_path, dest_path)
        with open(os.path.join(restructured_dir, 'val', 'labels', class_label, f"{image_id}.txt"), 'w') as f:
            f.write(f"{class_names.index(class_label)} 0.5 0.5 1.0 1.0\n")


for c in class_names:
    p = Augmentor.Pipeline(os.path.join(restructured_dir, 'train', 'images', c))
    p.rotate(probability=0.7, max_left_rotation=10, max_right_rotation=10)
    p.sample(500)
```

```
Initialised with 5385 image(s) found.
Output directory set to /content/ham10000_restructured/train/images/nv/output.Processing <PIL.JpegImagePlugin.JpegImageFile image mode=RGB siz
Initialised with 873 image(s) found.
Output directory set to /content/ham10000_restructured/train/images/mel/output.Processing <PIL.Image.Image image mode=RGB size=600x450 at 0x7E
Initialised with 874 image(s) found.
Output directory set to /content/ham10000_restructured/train/images/bkl/output.Processing <PIL.Image.Image image mode=RGB size=600x450 at 0x7E
Initialised with 414 image(s) found.
Output directory set to /content/ham10000_restructured/train/images/bcc/output.Processing <PIL.Image.Image image mode=RGB size=600x450 at 0x7E
Initialised with 263 image(s) found.
Output directory set to /content/ham10000_restructured/train/images/akiec/output.Processing <PIL.Image.Image image mode=RGB size=600x450 at 0x
Initialised with 112 image(s) found.
Output directory set to /content/ham10000_restructured/train/images/vasc/output.Processing <PIL.Image.Image image mode=RGB size=600x450 at 0x7
Initialised with 91 image(s) found.
Output directory set to /content/ham10000_restructured/train/images/df/output.Processing <PIL.JpegImagePlugin.JpegImageFile image mode=RGB siz
```

```
train_ds = tf.keras.preprocessing.image_dataset_from_directory(
    os.path.join(restructured_dir, 'train', 'images'), batch_size=16, image_size=(128, 128),
    label_mode='categorical', seed=123, subset="training", validation_split=0.2
)
val_ds = tf.keras.preprocessing.image_dataset_from_directory(
    os.path.join(restructured_dir, 'val', 'images'), batch_size=16, image_size=(128, 128),
```

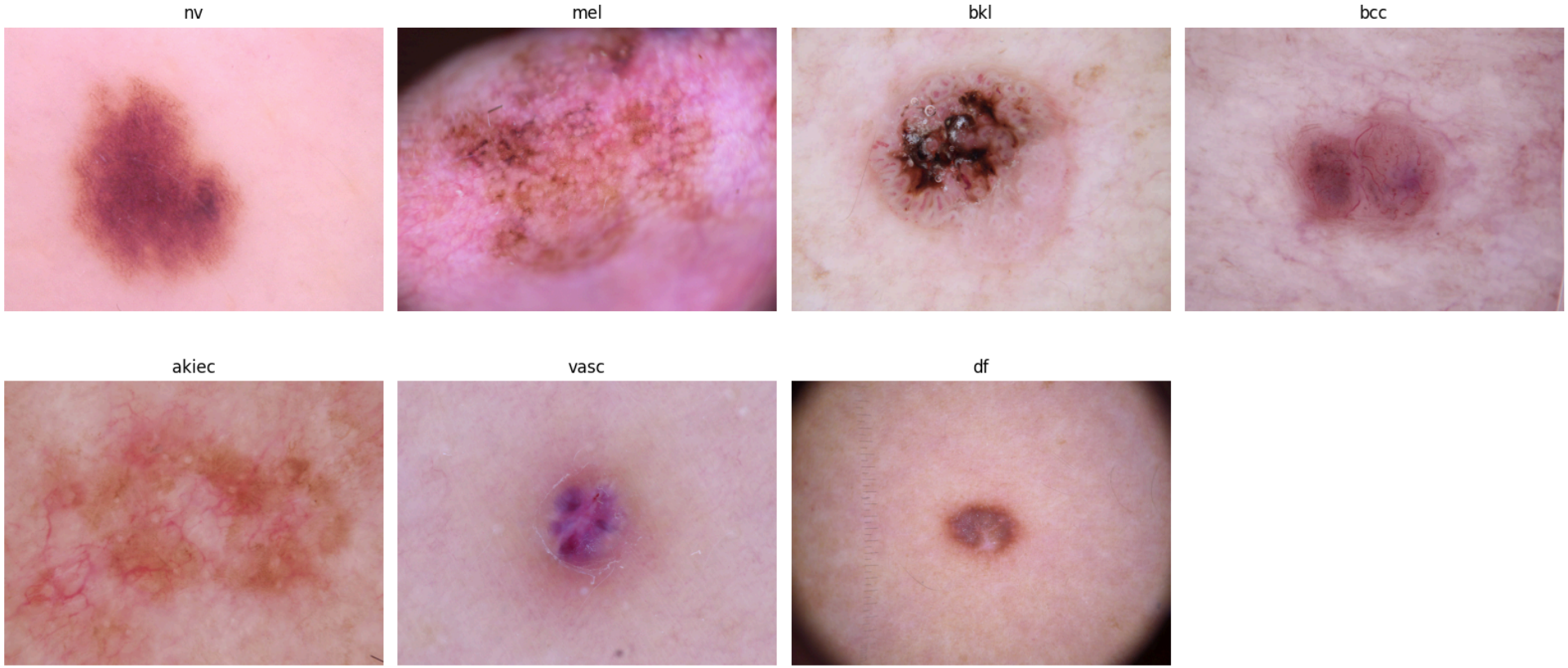
```
    label_mode='categorical', seed=123, subset="validation", validation_split=0.2
)
```

```
Found 11512 files belonging to 7 classes.
Using 9210 files for training.
Found 2003 files belonging to 7 classes.
Using 400 files for validation.
```

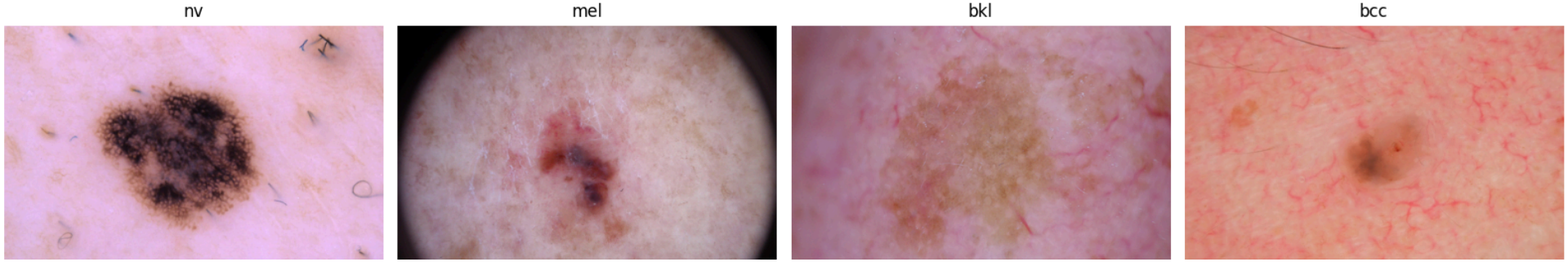
```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg


def show_one_image_per_class(folder_path, class_names, title="Folder"):
    plt.figure(figsize=(16, 8))
    for i, class_name in enumerate(class_names):
        class_folder = os.path.join(folder_path, class_name)
        image_list = os.listdir(class_folder)
        if image_list:
            image_path = os.path.join(class_folder, image_list[0])
            img = mpimg.imread(image_path)
            plt.subplot(2, (len(class_names)+1)//2, i+1)
            plt.imshow(img)
            plt.title(class_name)
            plt.axis('off')
    plt.suptitle(f' One Image from Each Class in {title}', fontsize=16)
    plt.tight_layout()
    plt.show()
train_path = os.path.join(restructured_dir, 'train', 'images')
train_classes = class_names

val_path = os.path.join(restructured_dir, 'val', 'images')
val_classes = class_names

show_one_image_per_class(train_path, train_classes, title="Train")
show_one_image_per_class(val_path, val_classes, title="Validation")
```

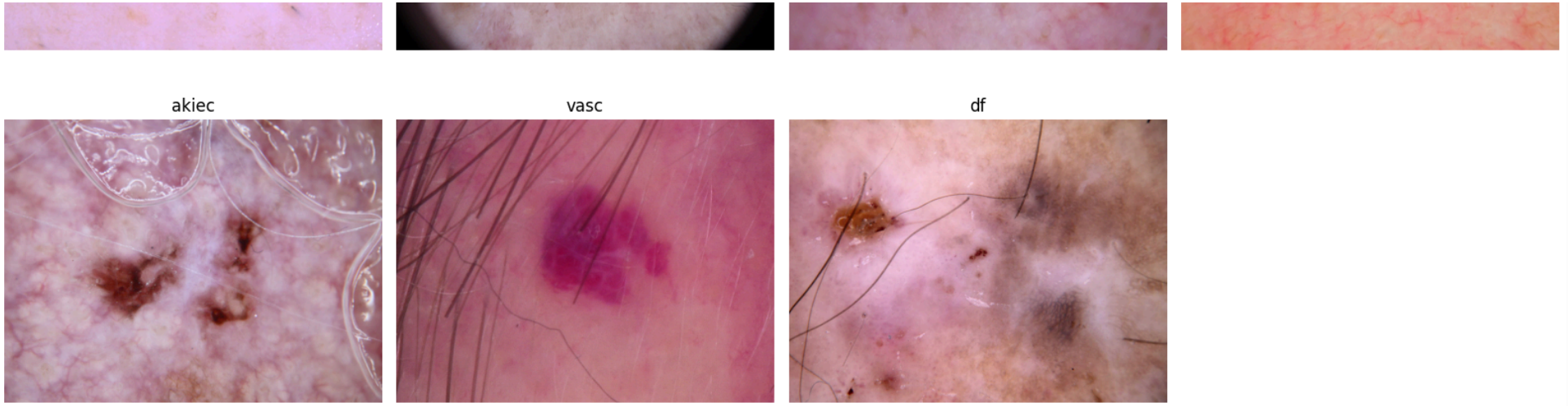
```
<ipython-input-13-55b07c51f75f>:18: UserWarning: Glyph 128444 (\N{FRAME WITH PICTURE}) missing from font(s) DejaVu Sans.
  plt.tight_layout()
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 128444 (\N{FRAME WITH PICTURE}) missing from fon
  fig.canvas.print_figure(bytes_io, **kw)
```

 One Image from Each Class in Train



 One Image from Each Class in Validation






```
AUTOTUNE = tf.data.AUTOTUNE
train_ds = train_ds.cache().shuffle(1000).prefetch(buffer_size=AUTOTUNE)
val_ds = val_ds.cache().prefetch(buffer_size=AUTOTUNE)


checkpoint_cnn = tf.keras.callbacks.ModelCheckpoint("cnn_model.h5", monitor="val_accuracy", save_best_only=True, verbose=1)
earlystop_cnn = tf.keras.callbacks.EarlyStopping(monitor="val_accuracy", patience=5, verbose=1)
checkpoint_resnet = tf.keras.callbacks.ModelCheckpoint("resnet_model.h5", monitor="val_accuracy", save_best_only=True, verbose=1)
earlystop_resnet = tf.keras.callbacks.EarlyStopping(monitor="val_accuracy", patience=5, verbose=1)


def create_cnn_model():
    model = models.Sequential([
        layers.Rescaling(1./255, input_shape=(128, 128, 3)),
        layers.Conv2D(32, (3, 3), activation='relu'),
        layers.MaxPooling2D((2, 2)),
        layers.Conv2D(64, (3, 3), activation='relu'),
        layers.MaxPooling2D((2, 2)),
        layers.Conv2D(128, (3, 3), activation='relu'),
        layers.MaxPooling2D((2, 2)),
        layers.Dropout(0.5),
```

```
        layers.Flatten(),
        layers.Dense(128, activation='relu'),
        layers.Dropout(0.25),
        layers.Dense(len(class_names), activation='softmax')
    ])
    model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
    return model


import os
os.environ["CUDA_VISIBLE_DEVICES"] = "-1"


import os

if tf.config.list_physical_devices('GPU'):
    print("GPU found, using the GPU for training")
else:
    print("No GPU found, using the CPU for training")

cnn_model = create_cnn_model()
cnn_history = cnn_model.fit(train_ds, validation_data=val_ds, epochs=15, callbacks=[checkpoint_cnn, earlystop_cnn])
```

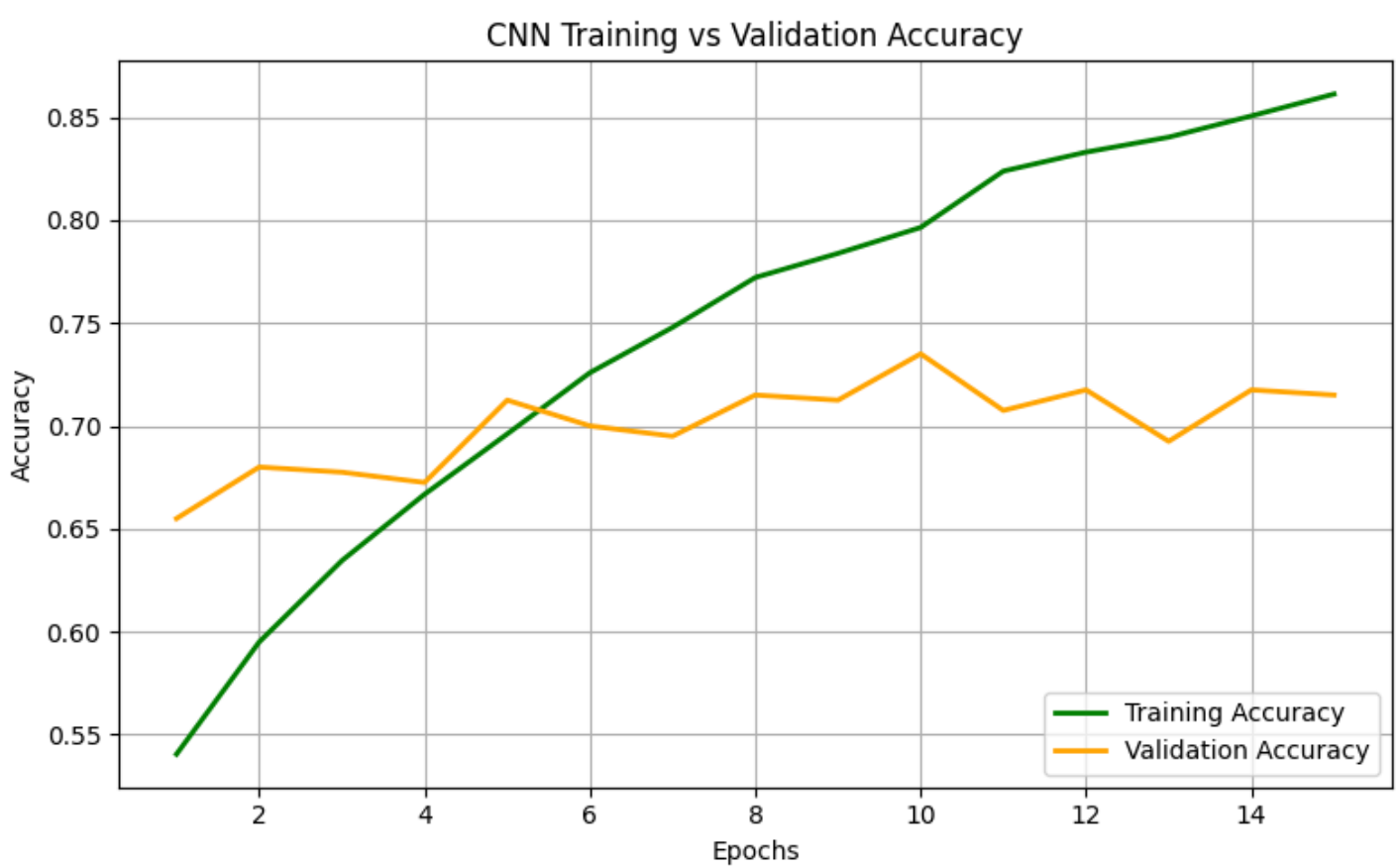
```
576/576 ━━━━━━━━━━━━━━━━━━━━ 284s 494ms/step - accuracy: 0.5830 - loss: 1.0989 - val_accuracy: 0.6800 - val_loss: 0.8461
Epoch 3/15
576/576 ━━━━━━━━━━━━━━━━━━━━ 0s 505ms/step - accuracy: 0.6381 - loss: 0.9626
Epoch 3: val_accuracy did not improve from 0.68000
576/576 ━━━━━━━━━━━━━━━━━━━━ 332s 511ms/step - accuracy: 0.6381 - loss: 0.9626 - val_accuracy: 0.6775 - val_loss: 0.7966
Epoch 4/15
576/576 ━━━━━━━━━━━━━━━━━━━━ 0s 488ms/step - accuracy: 0.6643 - loss: 0.8912
Epoch 4: val_accuracy did not improve from 0.68000
576/576 ━━━━━━━━━━━━━━━━━━━━ 286s 497ms/step - accuracy: 0.6643 - loss: 0.8912 - val_accuracy: 0.6725 - val_loss: 0.8211
```

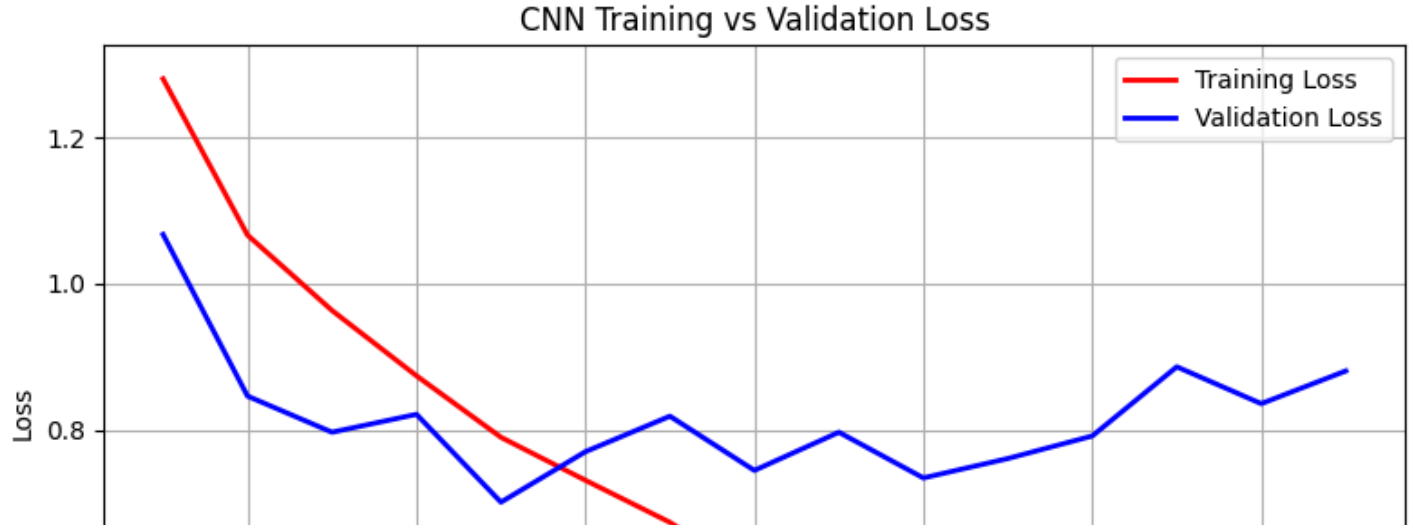
```
576/576 ━━━━━━━━━━━━━━━━━━━━ 290s 504ms/step - accuracy: 0.7412 - loss: 0.6777 - val_accuracy: 0.6950 - val_loss: 0.8186
```

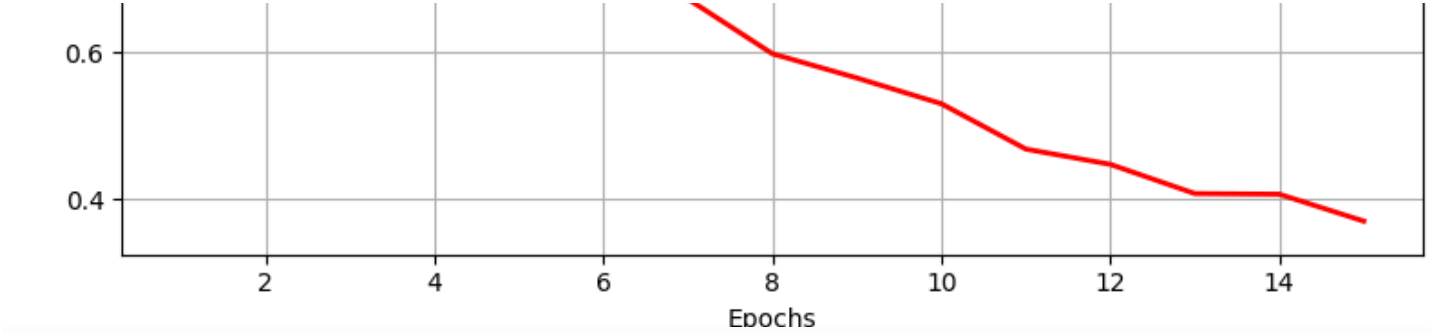
```
Epoch 8/15
576/576 ━━━━━━━━━━━━━━━━━━━━ 0s 494ms/step - accuracy: 0.7694 - loss: 0.6062
Epoch 8: val_accuracy improved from 0.71250 to 0.71500, saving model to cnn_model.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is consider
576/576 ━━━━━━━━━━━━━━━━━━━━ 290s 504ms/step - accuracy: 0.7694 - loss: 0.6062 - val_accuracy: 0.7150 - val_loss: 0.7443
Epoch 9/15
576/576 ━━━━━━━━━━━━━━━━━━━━ 0s 489ms/step - accuracy: 0.7919 - loss: 0.5497
Epoch 9: val_accuracy did not improve from 0.71500
576/576 ━━━━━━━━━━━━━━━━━━━━ 318s 497ms/step - accuracy: 0.7919 - loss: 0.5497 - val_accuracy: 0.7125 - val_loss: 0.7966
Epoch 10/15
576/576 ━━━━━━━━━━━━━━━━━━━━ 0s 489ms/step - accuracy: 0.8017 - loss: 0.5154
Epoch 10: val_accuracy improved from 0.71500 to 0.73500, saving model to cnn_model.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is consider
576/576 ━━━━━━━━━━━━━━━━━━━━ 284s 494ms/step - accuracy: 0.8017 - loss: 0.5154 - val_accuracy: 0.7350 - val_loss: 0.7340
Epoch 11/15
576/576 ━━━━━━━━━━━━━━━━━━━━ 0s 500ms/step - accuracy: 0.8340 - loss: 0.4490
Epoch 11: val_accuracy did not improve from 0.73500
576/576 ━━━━━━━━━━━━━━━━━━━━ 292s 506ms/step - accuracy: 0.8340 - loss: 0.4491 - val_accuracy: 0.7075 - val_loss: 0.7606
Epoch 12/15
576/576 ━━━━━━━━━━━━━━━━━━━━ 0s 491ms/step - accuracy: 0.8309 - loss: 0.4458
Epoch 12: val_accuracy did not improve from 0.73500
576/576 ━━━━━━━━━━━━━━━━━━━━ 317s 499ms/step - accuracy: 0.8309 - loss: 0.4458 - val_accuracy: 0.7175 - val_loss: 0.7915
Epoch 13/15
576/576 ━━━━━━━━━━━━━━━━━━━━ 0s 488ms/step - accuracy: 0.8438 - loss: 0.4005
Epoch 13: val_accuracy did not improve from 0.73500
576/576 ━━━━━━━━━━━━━━━━━━━━ 284s 493ms/step - accuracy: 0.8438 - loss: 0.4005 - val_accuracy: 0.6925 - val_loss: 0.8862
Epoch 14/15
576/576 ━━━━━━━━━━━━━━━━━━━━ 0s 489ms/step - accuracy: 0.8538 - loss: 0.3963
Epoch 14: val_accuracy did not improve from 0.73500
576/576 ━━━━━━━━━━━━━━━━━━━━ 285s 494ms/step - accuracy: 0.8538 - loss: 0.3963 - val_accuracy: 0.7175 - val_loss: 0.8357
Epoch 15/15
576/576 ━━━━━━━━━━━━━━━━━━━━ 0s 490ms/step - accuracy: 0.8612 - loss: 0.3674
Epoch 15: val_accuracy did not improve from 0.73500
576/576 ━━━━━━━━━━━━━━━━━━━━ 324s 498ms/step - accuracy: 0.8612 - loss: 0.3674 - val_accuracy: 0.7150 - val_loss: 0.8805
Epoch 15: early stopping
```

```
import matplotlib.pyplot as plt

train_acc = cnn_history.history['accuracy']
val_acc = cnn_history.history['val_accuracy']
epochs = range(1, len(train_acc) + 1)

plt.figure(figsize=(8, 5))
```

```
plt.plot(epochs, train_acc, 'g-', label='Training Accuracy', linewidth=2)
plt.plot(epochs, val_acc, 'orange', label='Validation Accuracy', linewidth=2)
plt.title('CNN Training vs Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend(loc='lower right')
plt.grid(True)
plt.tight_layout()
plt.show()

train_loss = cnn_history.history['loss']
val_loss = cnn_history.history['val_loss']

plt.figure(figsize=(8, 5))
plt.plot(epochs, train_loss, 'r-', label='Training Loss', linewidth=2)
plt.plot(epochs, val_loss, 'blue', label='Validation Loss', linewidth=2)
plt.title('CNN Training vs Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend(loc='upper right')
plt.grid(True)
plt.tight_layout()
plt.show()
```

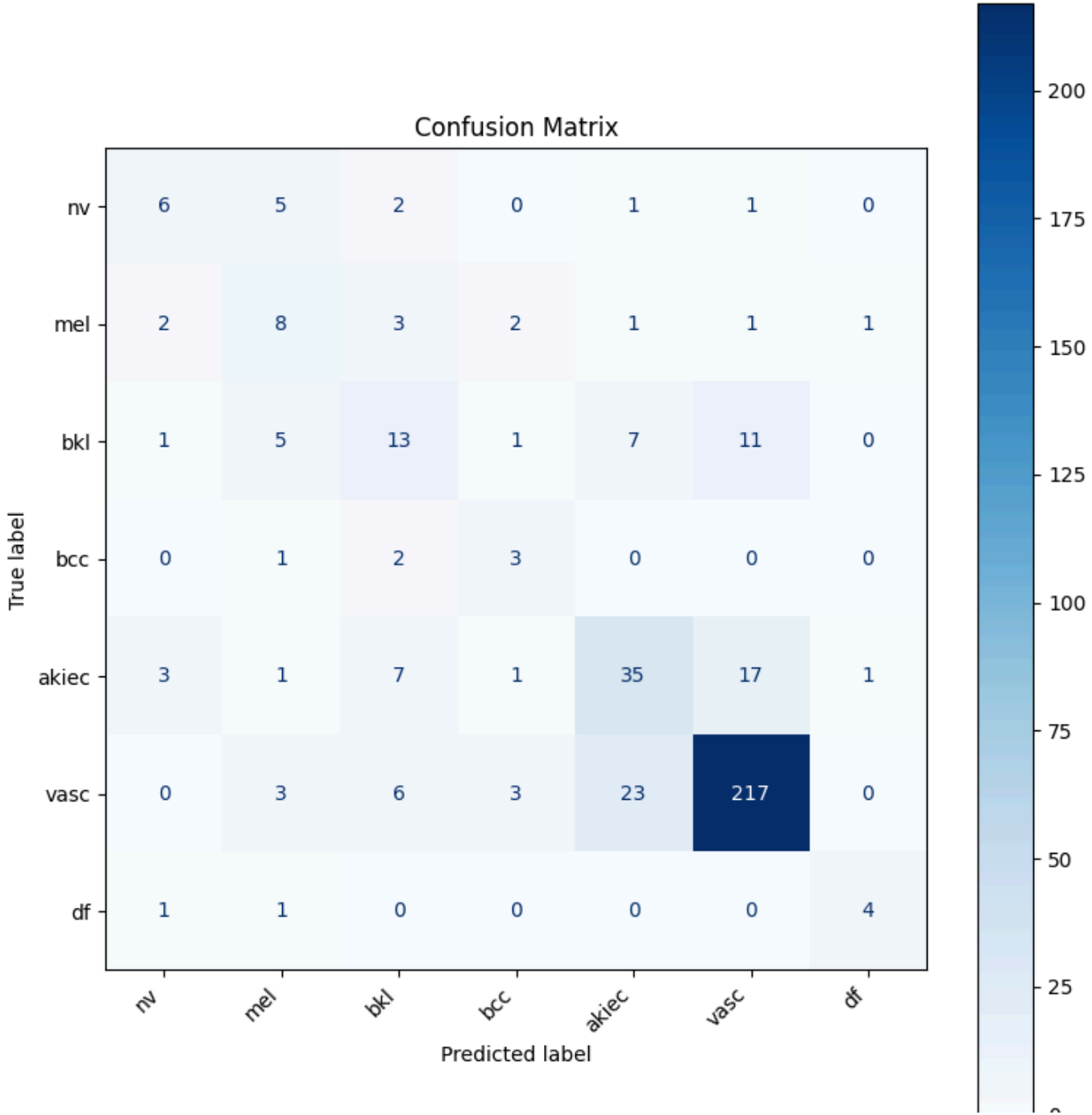

CNN Training vs Validation Accuracy
Accuracy
Epochs
Training Accuracy
Validation Accuracy
0.85
0.80
0.75
0.70
0.65
0.60
0.55
2
4
6
8
10
12
14


CNN Training vs Validation Loss
Loss
Training Loss
Validation Loss
1.2
1.0
0.8




```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

test_ds = val_ds

y_probs = cnn_model.predict(test_ds)
y_pred = np.argmax(y_probs, axis=1)

y_true_list = []
for X_batch, y_batch in test_ds:
    if y_batch.ndim > 1:
        y_true_list.append(np.argmax(y_batch, axis=1))
    else:
        y_true_list.append(y_batch)
y_true = np.concatenate(y_true_list, axis=0)

cm = confusion_matrix(y_true, y_pred)

disp = ConfusionMatrixDisplay(confusion_matrix=cm,
                              display_labels=class_names)
fig, ax = plt.subplots(figsize=(8, 8))
disp.plot(ax=ax, cmap='Blues', colorbar=True)
plt.title('Confusion Matrix')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

```
25/25 ━━━━━━━━━━━━━━━━━━━━ 3s 116ms/step
```



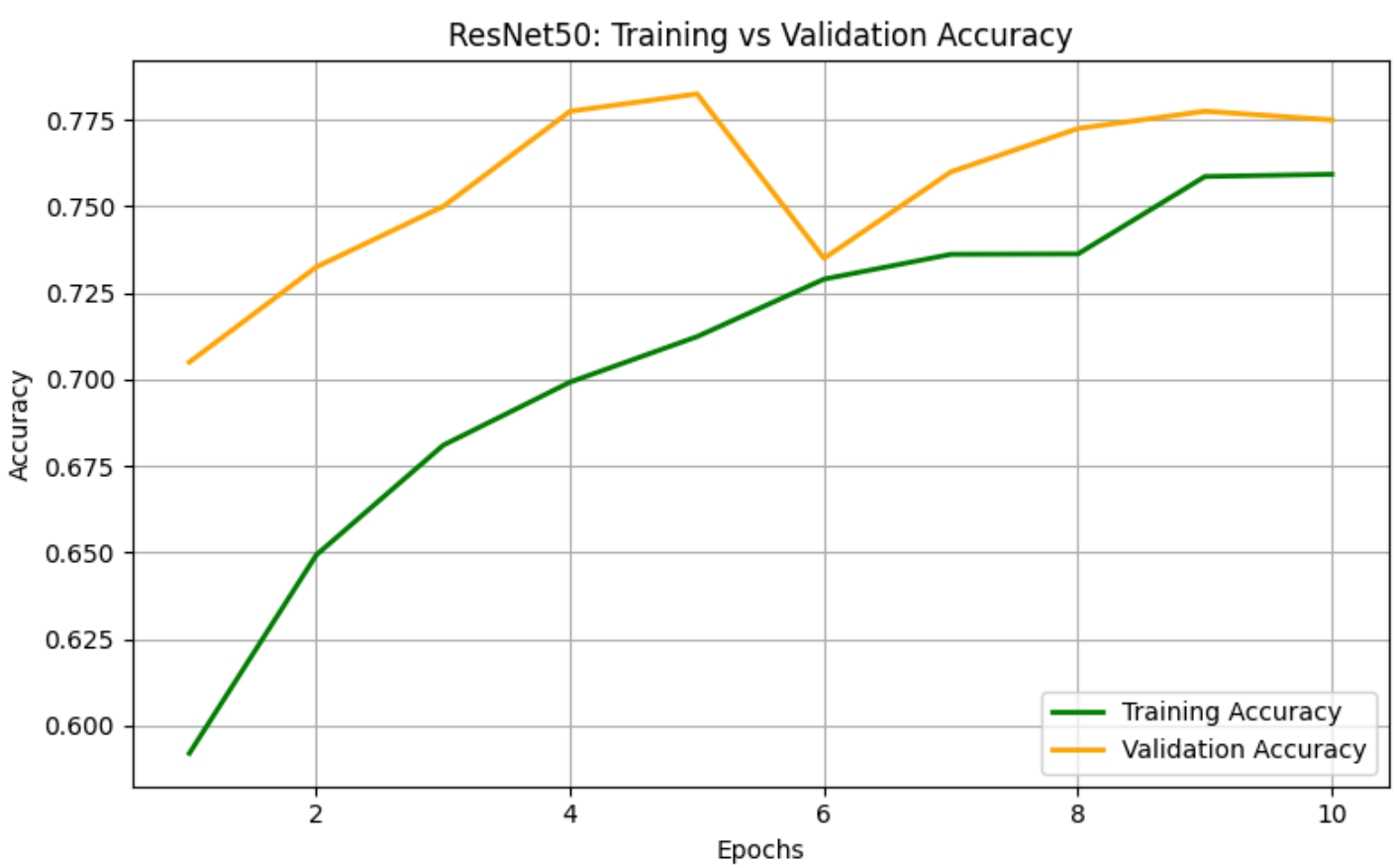
Start coding or generate with AI.

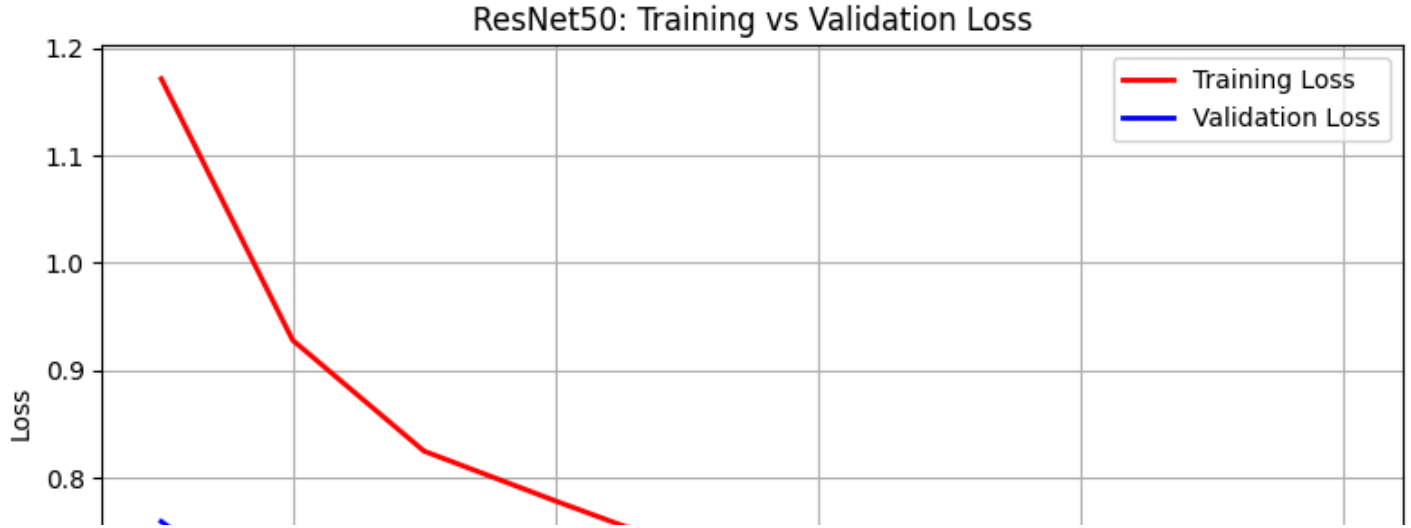
```
from tensorflow.keras.applications.resnet50 import preprocess_input, ResNet50

def create_resnet_model():

    base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(128, 128, 3))
    base_model.trainable = False

    model = models.Sequential([

        layers.Lambda(preprocess_input, input_shape=(128, 128, 3)),
        base_model,
        layers.GlobalAveragePooling2D(),
        layers.Dense(128, activation='relu'),
        layers.Dropout(0.5),
        layers.Dense(len(class_names), activation='softmax')
    ])


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

    return model


resnet_model = create_resnet_model()
resnet_history = resnet_model.fit(train_ds, validation_data=val_ds, epochs=15, callbacks=[checkpoint_resnet, earlystop_resnet])
```

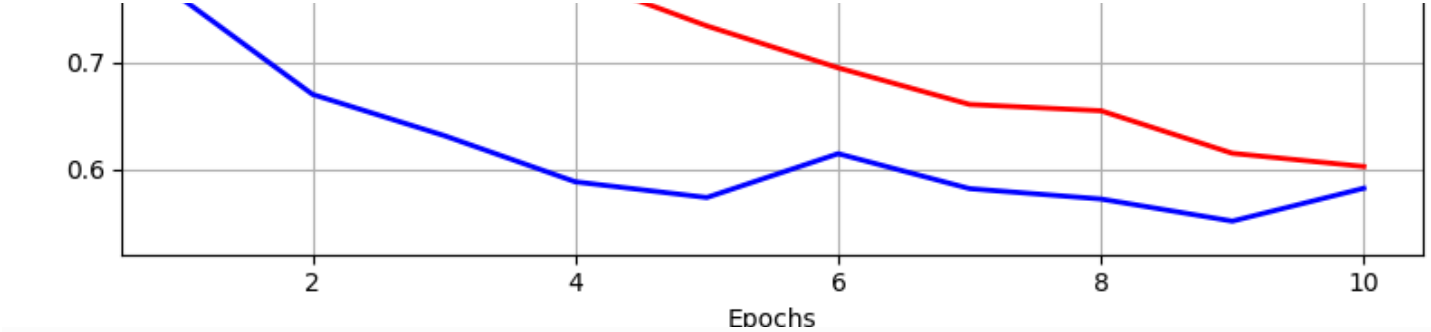
```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5
94765736/94765736 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
Epoch 1/15
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/lambda_layer.py:65: UserWarning: Do not pass an `input_shape`/`input_dim` argume
  super().__init__(**kwargs)
576/576 ━━━━━━━━━━━━━━━━━━━━ 0s 906ms/step - accuracy: 0.5415 - loss: 1.4260
Epoch 1: val_accuracy improved from -inf to 0.70500, saving model to resnet_model.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered
576/576 ━━━━━━━━━━━━━━━━━━━━ 559s 954ms/step - accuracy: 0.5416 - loss: 1.4255 - val_accuracy: 0.7050 - val_loss: 0.7594
```

```
Epoch 2/15
576/576 ━━━━━━━━━━━━━━━━━━━━ 0s 911ms/step - accuracy: 0.6432 - loss: 0.9510
Epoch 2: val_accuracy improved from 0.70500 to 0.73250, saving model to resnet_model.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered
576/576 ━━━━━━━━━━━━━━━━━━━━ 549s 953ms/step - accuracy: 0.6433 - loss: 0.9510 - val_accuracy: 0.7325 - val_loss: 0.6699
Epoch 3/15
576/576 ━━━━━━━━━━━━━━━━━━━━ 0s 906ms/step - accuracy: 0.6830 - loss: 0.8185
Epoch 3: val_accuracy improved from 0.73250 to 0.75000, saving model to resnet_model.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered
576/576 ━━━━━━━━━━━━━━━━━━━━ 558s 946ms/step - accuracy: 0.6830 - loss: 0.8185 - val_accuracy: 0.7500 - val_loss: 0.6317
Epoch 4/15
576/576 ━━━━━━━━━━━━━━━━━━━━ 0s 905ms/step - accuracy: 0.7023 - loss: 0.7689
Epoch 4: val_accuracy improved from 0.75000 to 0.77750, saving model to resnet_model.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered
576/576 ━━━━━━━━━━━━━━━━━━━━ 544s 944ms/step - accuracy: 0.7023 - loss: 0.7689 - val_accuracy: 0.7775 - val_loss: 0.5887
Epoch 5/15
576/576 ━━━━━━━━━━━━━━━━━━━━ 0s 908ms/step - accuracy: 0.7159 - loss: 0.7173
Epoch 5: val_accuracy improved from 0.77750 to 0.78250, saving model to resnet_model.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered
576/576 ━━━━━━━━━━━━━━━━━━━━ 565s 949ms/step - accuracy: 0.7159 - loss: 0.7173 - val_accuracy: 0.7825 - val_loss: 0.5741
Epoch 6/15
576/576 ━━━━━━━━━━━━━━━━━━━━ 0s 899ms/step - accuracy: 0.7275 - loss: 0.7106
Epoch 6: val_accuracy did not improve from 0.78250
576/576 ━━━━━━━━━━━━━━━━━━━━ 556s 939ms/step - accuracy: 0.7275 - loss: 0.7105 - val_accuracy: 0.7350 - val_loss: 0.6148
Epoch 7/15
576/576 ━━━━━━━━━━━━━━━━━━━━ 0s 901ms/step - accuracy: 0.7347 - loss: 0.6490
Epoch 7: val_accuracy did not improve from 0.78250
576/576 ━━━━━━━━━━━━━━━━━━━━ 542s 941ms/step - accuracy: 0.7347 - loss: 0.6490 - val_accuracy: 0.7600 - val_loss: 0.5823
Epoch 8/15
576/576 ━━━━━━━━━━━━━━━━━━━━ 0s 896ms/step - accuracy: 0.7406 - loss: 0.6502
Epoch 8: val_accuracy did not improve from 0.78250
576/576 ━━━━━━━━━━━━━━━━━━━━ 557s 968ms/step - accuracy: 0.7406 - loss: 0.6502 - val_accuracy: 0.7725 - val_loss: 0.5727
Epoch 9/15
576/576 ━━━━━━━━━━━━━━━━━━━━ 0s 899ms/step - accuracy: 0.7556 - loss: 0.6139
Epoch 9: val_accuracy did not improve from 0.78250
576/576 ━━━━━━━━━━━━━━━━━━━━ 559s 970ms/step - accuracy: 0.7556 - loss: 0.6139 - val_accuracy: 0.7775 - val_loss: 0.5521
Epoch 10/15
576/576 ━━━━━━━━━━━━━━━━━━━━ 0s 902ms/step - accuracy: 0.7629 - loss: 0.5986
Epoch 10: val_accuracy did not improve from 0.78250
576/576 ━━━━━━━━━━━━━━━━━━━━ 542s 940ms/step - accuracy: 0.7629 - loss: 0.5986 - val_accuracy: 0.7750 - val_loss: 0.5827
Epoch 10: early stopping
```

```
import matplotlib.pyplot as plt

train_acc = resnet_history.history['accuracy']
val_acc = resnet_history.history['val_accuracy']
train_loss = resnet_history.history['loss']
val_loss = resnet_history.history['val_loss']
epochs = range(1, len(train_acc) + 1)

plt.figure(figsize=(8, 5))
plt.plot(epochs, train_acc, 'green', label='Training Accuracy', linewidth=2)
plt.plot(epochs, val_acc, 'orange', label='Validation Accuracy', linewidth=2)
plt.title('ResNet50: Training vs Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend(loc='lower right')
plt.grid(True)
plt.tight_layout()
plt.show()

plt.figure(figsize=(8, 5))
plt.plot(epochs, train_loss, 'red', label='Training Loss', linewidth=2)
plt.plot(epochs, val_loss, 'blue', label='Validation Loss', linewidth=2)
plt.title('ResNet50: Training vs Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend(loc='upper right')
plt.grid(True)
plt.tight_layout()
plt.show()
```

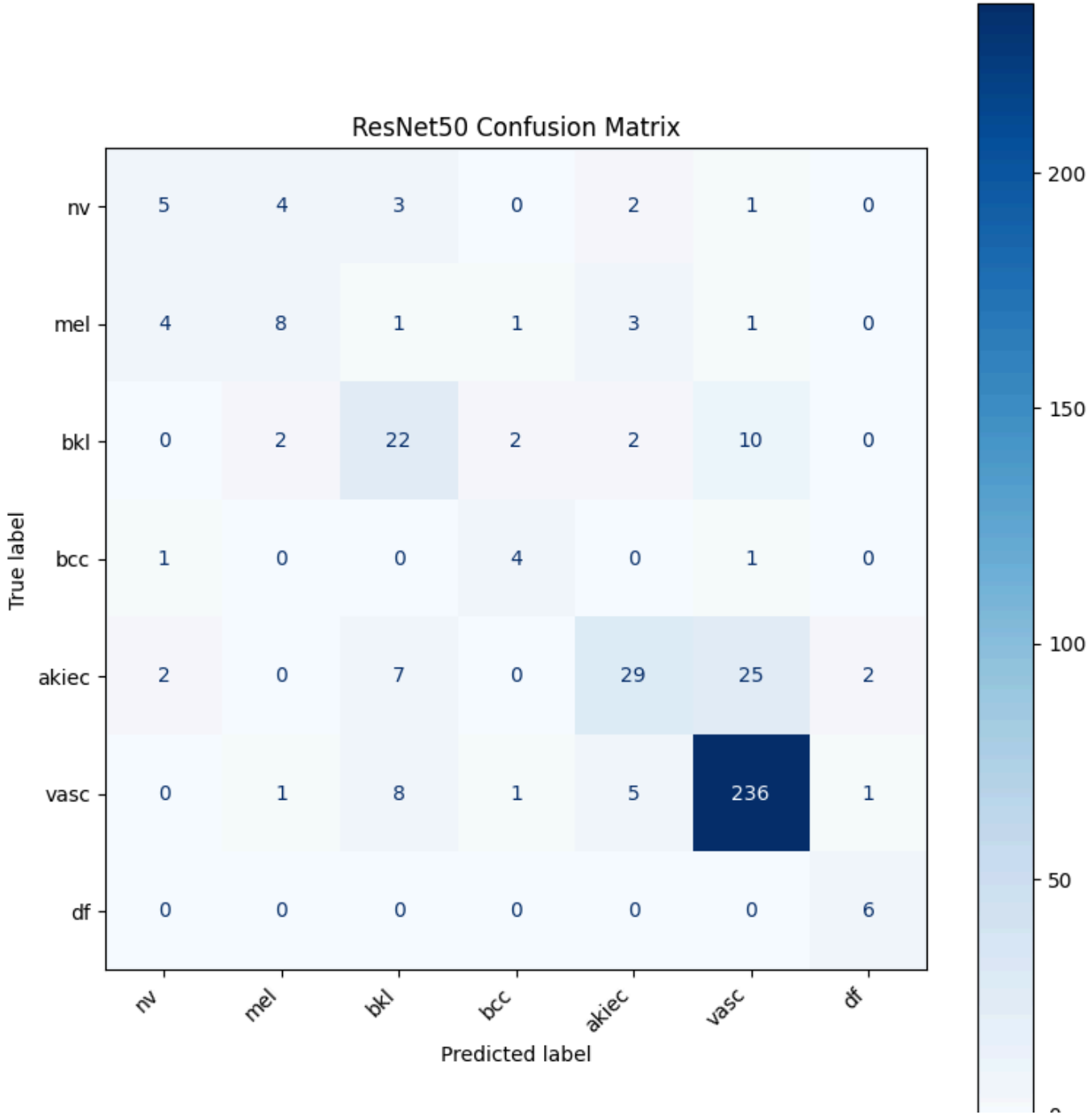

ResNet50: Training vs Validation Accuracy
Accuracy
Epochs
Training Accuracy
Validation Accuracy
0.775
0.750
0.725
0.700
0.675
0.650
0.625
0.600
2
4
6
8
10


ResNet50: Training vs Validation Loss
Loss
Training Loss
Validation Loss
1.2
1.1
1.0
0.9
0.8




```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

test_ds = val_ds
y_probs = resnet_model.predict(test_ds)
y_pred = np.argmax(y_probs, axis=1)

y_true_list = []
for X_batch, y_batch in test_ds:
    if y_batch.ndim > 1:
        y_true_list.append(np.argmax(y_batch, axis=1))
    else:
        y_true_list.append(y_batch)
y_true = np.concatenate(y_true_list, axis=0)

cm = confusion_matrix(y_true, y_pred)

disp = ConfusionMatrixDisplay(confusion_matrix=cm,
                              display_labels=class_names)
fig, ax = plt.subplots(figsize=(8, 8))
disp.plot(ax=ax, cmap='Blues', colorbar=True)
plt.title('ResNet50 Confusion Matrix')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

```
25/25 ━━━━━━━━━━━━━━━━━━━━ 25s 857ms/step
```



```
yolo_model = YOLO('/content/drive/MyDrive/yolov8m.pt')
data_yaml = f"""
train: {os.path.join(restructured_dir, 'train', 'images')}
val: {os.path.join(restructured_dir, 'val', 'images')}
nc: {len(class_names)}
names: {class_names}
"""
with open(os.path.join(restructured_dir, 'data.yaml'), 'w') as f:
    f.write(data_yaml)
yolo_model.train(data=os.path.join(restructured_dir, 'data.yaml'), epochs=3, imgsz=64, batch=32)
```

```
Ultralytics 8.3.114 🚀 Python-3.11.12 torch-2.6.0+cu124 CPU (Intel Xeon 2.20GHz)
engine/trainer: task=detect, mode=train, model=/content/drive/MyDrive/yolov8m.pt, data=/content/ham10000_restructured/data.yaml, epochs=3,
Downloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/Ultralytics/Arial.ttf'...
100%|██████████| 755k/755k [00:00<00:00, 10.6MB/s]Overriding model.yaml nc=80 with nc=7

                   from  n    params  module                                       arguments
  0                  -1  1      1392  ultralytics.nn.modules.conv.Conv             [3, 48, 3, 2]
  1                  -1  1     41664  ultralytics.nn.modules.conv.Conv             [48, 96, 3, 2]
  2                  -1  2    111360  ultralytics.nn.modules.block.C2f             [96, 96, 2, True]
  3                  -1  1    166272  ultralytics.nn.modules.conv.Conv             [96, 192, 3, 2]
  4                  -1  4    813312  ultralytics.nn.modules.block.C2f             [192, 192, 4, True]
  5                  -1  1    664320  ultralytics.nn.modules.conv.Conv             [192, 384, 3, 2]

  6                  -1  4   3248640  ultralytics.nn.modules.block.C2f             [384, 384, 4, True]
  7                  -1  1   1991808  ultralytics.nn.modules.conv.Conv             [384, 576, 3, 2]
  8                  -1  2   3985920  ultralytics.nn.modules.block.C2f             [576, 576, 2, True]
  9                  -1  1    831168  ultralytics.nn.modules.block.SPPF            [576, 576, 5]
 10                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 11             [-1, 6]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 12                  -1  2   1993728  ultralytics.nn.modules.block.C2f             [960, 384, 2]
 13                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 14             [-1, 4]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 15                  -1  2    517632  ultralytics.nn.modules.block.C2f             [576, 192, 2]
 16                  -1  1    332160  ultralytics.nn.modules.conv.Conv             [192, 192, 3, 2]
 17            [-1, 12]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 18                  -1  2   1846272  ultralytics.nn.modules.block.C2f             [576, 384, 2]
 19                  -1  1   1327872  ultralytics.nn.modules.conv.Conv             [384, 384, 3, 2]
 20             [-1, 9]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 21                  -1  2   4207104  ultralytics.nn.modules.block.C2f             [960, 576, 2]
 22        [15, 18, 21]  1   3779749  ultralytics.nn.modules.head.Detect           [7, [192, 384, 576]]
```

```
Model summary: 169 layers, 25,860,373 parameters, 25,860,357 gradients, 79.1 GFLOPs

Transferred 469/475 items from pretrained weights
Freezing layer 'model.22.dfl.conv.weight'
train: Fast image access ✅ (ping: 0.1±0.0 ms, read: 170.3±300.2 MB/s, size: 217.8 KB)
train: Scanning /content/ham10000_restructured/train/labels/akiec... 8012 images, 3500 backgrounds, 0 corrupt: 100%|██████████| 11512/11512
train: New cache created: /content/ham10000_restructured/train/labels/akiec.cache
albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01, num_output_channels=3, method='weigh
val: Fast image access ✅ (ping: 0.0±0.0 ms, read: 73.1±7.9 MB/s, size: 288.9 KB)
val: Scanning /content/ham10000_restructured/val/labels/akiec... 2003 images, 0 backgrounds, 0 corrupt: 100%|██████████| 2003/2003 [00:05<0
val: New cache created: /content/ham10000_restructured/val/labels/akiec.cache
Plotting labels to runs/detect/train/labels.jpg...
optimizer: 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937' and determining best 'optimizer', 'lr0' and 'momentum' automati
optimizer: AdamW(lr=0.000909, momentum=0.9) with parameter groups 77 weight(decay=0.0), 84 weight(decay=0.0005), 83 bias(decay=0.0)
Image sizes 64 train, 64 val
Using 0 dataloader workers
Logging results to runs/detect/train
Starting training for 3 epochs...

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
        1/3         0G     0.4688      1.596      1.019         59         64: 100%|██████████| 360/360 [17:26<00:00,  2.91s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 32/32 [01:21<00:00,  2.56s/it]


      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
        2/3         0G     0.2255      1.051     0.9189         60         64: 100%|██████████| 360/360 [16:54<00:00,  2.82s/it]
```

```
print(yolo_model)
```

```
YOLO(
  (model): DetectionModel(
    (model): Sequential(
      (0): Conv(
        (conv): Conv2d(3, 48, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
        (bn): BatchNorm2d(48, eps=0.001, momentum=0.03, affine=True, track_running_stats=True)
        (act): SiLU(inplace=True)
      )
      (1): Conv(
        (conv): Conv2d(48, 96, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
        (bn): BatchNorm2d(96, eps=0.001, momentum=0.03, affine=True, track_running_stats=True)
        (act): SiLU(inplace=True)
      )
      (2): C2f(
        (cv1): Conv(
```

```
      (conv): Conv2d(96, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn): BatchNorm2d(96, eps=0.001, momentum=0.03, affine=True, track_running_stats=True)
      (act): SiLU(inplace=True)
    )
    (cv2): Conv(
      (conv): Conv2d(192, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn): BatchNorm2d(96, eps=0.001, momentum=0.03, affine=True, track_running_stats=True)
      (act): SiLU(inplace=True)
    )
    (m): ModuleList(
      (0-1): 2 x Bottleneck(
        (cv1): Conv(
          (conv): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
          (bn): BatchNorm2d(48, eps=0.001, momentum=0.03, affine=True, track_running_stats=True)
          (act): SiLU(inplace=True)
        )
        (cv2): Conv(
          (conv): Conv2d(48, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
          (bn): BatchNorm2d(48, eps=0.001, momentum=0.03, affine=True, track_running_stats=True)
          (act): SiLU(inplace=True)
        )
      )
    )
  )
  (3): Conv(
    (conv): Conv2d(96, 192, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
    (bn): BatchNorm2d(192, eps=0.001, momentum=0.03, affine=True, track_running_stats=True)
    (act): SiLU(inplace=True)
  )
  (4): C2f(
    (cv1): Conv(
      (conv): Conv2d(192, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn): BatchNorm2d(192, eps=0.001, momentum=0.03, affine=True, track_running_stats=True)
      (act): SiLU(inplace=True)
    )
    (cv2): Conv(
      (conv): Conv2d(576, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn): BatchNorm2d(192, eps=0.001, momentum=0.03, affine=True, track_running_stats=True)
      (act): SiLU(inplace=True)
    )
    (m): ModuleList(
      (0-3): 4 x Bottleneck(
        (cv1): Conv(
```

```
    def evaluate_classification(model, val_ds, model_name):
        val_labels = np.concatenate([y for _, y in val_ds], axis=0)
        predictions = model.predict(val_ds)
        y_true = np.argmax(val_labels, axis=1)
        y_pred = np.argmax(predictions, axis=1)
        print(f"\n{model_name} Classification Report:")
        print(classification_report(y_true, y_pred, target_names=[lesion_type_dict[c]['name'] for c in class_names]))
        severities = [lesion_type_dict[class_names[p]]['severity'] for p in y_pred]
        print(f"{model_name} Severity Distribution:")
        print(pd.Series(severities).value_counts())


    def evaluate_yolo(model, val_dir, metadata):
        val_images_dir = os.path.join(val_dir, 'images')
        print(f"Looking for images in: {val_images_dir}")
        val_images = glob.glob(os.path.join(val_images_dir, '*', '*.jpg'))
        print(f"Found {len(val_images)} images in subdirectories")
        if not val_images:
            raise FileNotFoundError(f"No images found in {val_images_dir} or its subdirectories.")
        results = model.predict(source=val_images, save=False, imgsz=128)
        y_true, y_pred, sizes, severities = [], [], [], []
        pixel_to_mm = 0.1
        for result in results:
            image_id = os.path.basename(result.path).split('.')[0]
            true_label = metadata[metadata['image_id'] == image_id]['dx'].values[0]
            y_true.append(class_names.index(true_label))
            if len(result.boxes) > 0:
                pred_class = int(result.boxes.cls[0])
                confidence = float(result.boxes.conf[0])
                y_pred.append(pred_class)
                box = result.boxes.xywh[0]
                width, height = float(box[2]), float(box[3])
                size_mm2 = (width * pixel_to_mm) * (height * pixel_to_mm)
                sizes.append(size_mm2)
                severity = lesion_type_dict[class_names[pred_class]]['severity'] if confidence > 0.5 else 'Low'
                severities.append(severity)
            else:
                y_pred.append(-1)
                sizes.append(0)
                severities.append('Low')
        print("\nYOLOv8 Classification Report:")
        print(classification_report(y_true, [p if p != -1 else y_true[i] for i, p in enumerate(y_pred)], target_names=class_names))
```

```
    print(f"Average Size of Infected Area: {np.mean(sizes):.2f} mm²")
    print("YOLOv11 Severity Distribution:")
    print(pd.Series(severities).value_counts())


val_images = glob.glob(os.path.join(restructured_dir, 'val', 'images', '*', '*.jpg'))
print(f"Number of validation images found: {len(val_images)}")
print(f"Validation metadata size: {len(val_meta)}")
print(val_meta.head())
```

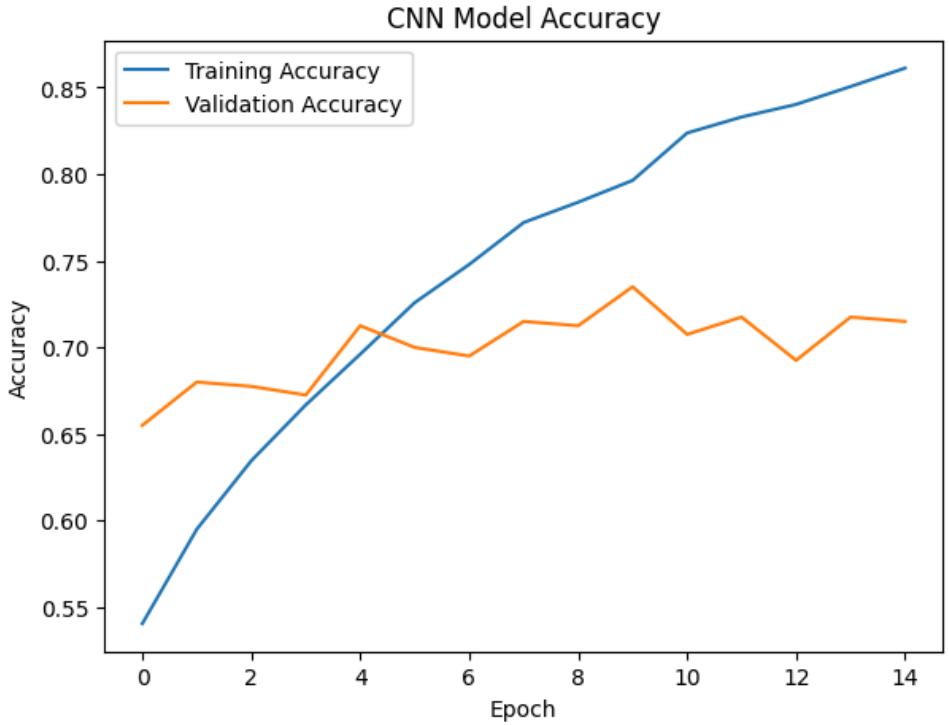
```
Number of validation images found: 2003
Validation metadata size: 2003
       lesion_id      image_id     dx    dx_type   age     sex  \
9725  HAM_0004376  ISIC_0024843  akiec      histo  70.0  female
6059  HAM_0003024  ISIC_0024768     nv  follow_up  35.0  female
4540  HAM_0001659  ISIC_0026564     nv  follow_up  35.0    male
3817  HAM_0004625  ISIC_0029346     nv  follow_up  40.0    male
7914  HAM_0000443  ISIC_0034271     nv      histo  35.0  female

         localization          cell_type  severity  cell_type_idx
9725             face  Actinic keratoses  Moderate              0
6059            trunk   Melanocytic nevi       Low              4
4540  lower extremity   Melanocytic nevi       Low              4
3817  upper extremity   Melanocytic nevi       Low              4
7914             back   Melanocytic nevi       Low              4
```

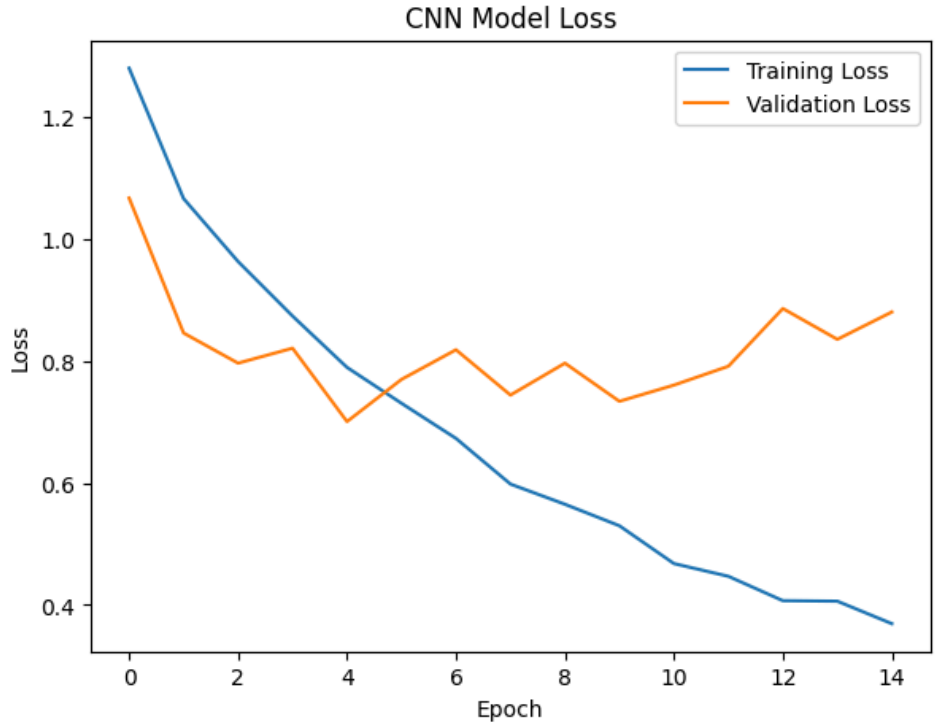
```
evaluate_classification(cnn_model, val_ds, "CNN")
evaluate_classification(resnet_model, val_ds, "ResNet")
```

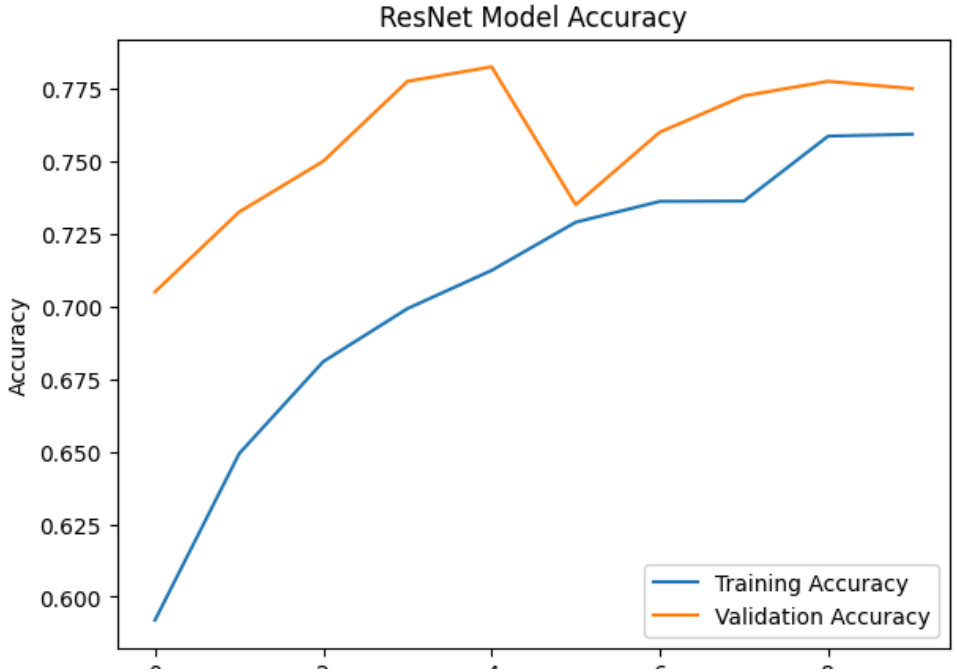
```
25/25 ━━━━━━━━━━━━━━━━━━━━ 5s 186ms/step

CNN Classification Report:
                              precision    recall  f1-score   support

             Melanocytic nevi       0.46      0.40      0.43        15
                     Melanoma       0.33      0.44      0.38        18
Benign keratosis-like lesions       0.39      0.34      0.37        38
         Basal cell carcinoma       0.30      0.50      0.38         6
            Actinic keratoses       0.52      0.54      0.53        65
             Vascular lesions       0.88      0.86      0.87       252
               Dermatofibroma       0.67      0.67      0.67         6

                     accuracy                           0.71       400
```

```
                      macro avg       0.51      0.54      0.52       400
                   weighted avg       0.72      0.71      0.72       400

CNN Severity Distribution:
Low         266
Moderate    100
High         34
Name: count, dtype: int64
25/25 ━━━━━━━━━━━━━━━━━━━━ 26s 1s/step

ResNet Classification Report:
                              precision    recall  f1-score   support

             Melanocytic nevi       0.42      0.33      0.37        15
                     Melanoma       0.53      0.44      0.48        18
Benign keratosis-like lesions       0.54      0.58      0.56        38
         Basal cell carcinoma       0.50      0.67      0.57         6
            Actinic keratoses       0.71      0.45      0.55        65
             Vascular lesions       0.86      0.94      0.90       252
               Dermatofibroma       0.67      1.00      0.80         6

                     accuracy                           0.78       400
                    macro avg       0.60      0.63      0.60       400
                 weighted avg       0.77      0.78      0.76       400

ResNet Severity Distribution:
Low         295
Moderate     82
High         23
Name: count, dtype: int64


def plot_history(history, title):
    plt.figure(figsize=(15, 5))
    plt.subplot(1, 2, 1)
    plt.plot(history.history['accuracy'], label='Training Accuracy')
    plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
    plt.title(f'{title} Accuracy')
    plt.xlabel('Epoch')
    plt.ylabel('Accuracy')
    plt.legend()
    plt.subplot(1, 2, 2)
    plt.plot(history.history['loss'], label='Training Loss')
    plt.plot(history.history['val_loss'], label='Validation Loss')
```

```
        plt.title(f'{title} Loss')
        plt.xlabel('Epoch')
        plt.ylabel('Loss')
        plt.legend()
        plt.show()


    def plot_yolo_predictions(model, val_dir, num_samples=5):
        val_images_dir = os.path.join(val_dir, 'images')
        val_images = glob.glob(os.path.join(val_images_dir, '*', '*.jpg'))
        print(f"Found {len(val_images)} images in subdirectories")
        if not val_images:
            raise FileNotFoundError(f"No images found in {val_images_dir} or its subdirectories.")
        results = model.predict(source=val_images, save=False, imgsz=128)
        plt.figure(figsize=(15, 10))
        for i, result in enumerate(results[:num_samples]):
            img = cv2.imread(result.path)
            img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
            plt.subplot(1, num_samples, i+1)
            plt.imshow(img)
            if len(result.boxes) > 0:
                box = result.boxes.xyxy[0].numpy()
                cv2.rectangle(img, (int(box[0]), int(box[1])), (int(box[2]), int(box[3])), (255, 0, 0), 2)
                label = f"{class_names[int(result.boxes.cls[0])]} ({result.boxes.conf[0]:.2f})"
                plt.title(label)
            plt.axis('off')
        plt.show()


    plot_history(cnn_history, "CNN Model")
    plot_history(resnet_history, "ResNet Model")
```


CNN Model Accuracy
Training Accuracy
Validation Accuracy
Accuracy
Epoch


CNN Model Loss
Training Loss
Validation Loss
Loss
Epoch


ResNet Model Accuracy
Training Accuracy
Validation Accuracy
Accuracy


ResNet Model Loss
Training Loss
Validation Loss
Loss


```
from ultralytics import YOLO
from PIL import Image
import matplotlib.pyplot as plt
import os
import numpy as np

IMAGE_PATHS = [
    "/content/ham10000_restructured/val/images/nv/ISIC_0024321.jpg",
    "/content/ham10000_restructured/val/images/bkl/ISIC_0024381.jpg",
    "/content/ham10000_restructured/val/images/df/ISIC_0024330.jpg",
    "/content/ham10000_restructured/val/images/vasc/ISIC_0026456.jpg",
    "/content/ham10000_restructured/val/images/bkl/ISIC_0024358.jpg",
]
MODEL_PATH  = "/content/drive/MyDrive/best.pt"

custom_labels = {
    0: "Nevus",
    1: "Melanoma",
    2: "Benign Keratosis",
    3: "Basal Cell Carcinoma",
    4: "Actinic Keratoses",
    5: "Vascular Lesion",
    6: "Dermatofibroma"
}

folder_to_label = {
    "nv":   "Nevus",
    "mel":  "Melanoma",
    "bkl":  "Benign Keratosis",
    "bcc":  "Basal Cell Carcinoma",
    "akiec":"Actinic Keratoses",
    "vasc": "Vascular Lesion",
```

```
        "df":   "Dermatofibroma"
    }

    model = YOLO(MODEL_PATH)

    results = model(IMAGE_PATHS)

    print("\n YOLO Skin Lesion Batch Results\n" + "-"*70)
    for i, (path, res) in enumerate(zip(IMAGE_PATHS, results), start=1):
        actual_folder = os.path.basename(os.path.dirname(path))
        actual_label  = folder_to_label.get(actual_folder, "Unknown")
        print(f"[Image {i}] {os.path.basename(path)}")
        print(f"  • Actual Class : {actual_label}")

        if res.boxes:
            for j, box in enumerate(res.boxes, start=1):
                cls_id = int(box.cls[0])
                conf   = float(box.conf[0])
                pred   = custom_labels.get(cls_id, f"Unknown({cls_id})")
                print(f"    Prediction {j}: {pred} (conf: {conf:.2f})")
        else:
            print("    No lesions detected.")
        print("-"*70)

    fig, axes = plt.subplots(1, 5, figsize=(20, 5))
    for ax, res in zip(axes, results):
        annotated = res.plot()

        if annotated.shape[2] == 3 and annotated.dtype == np.uint8:
            annotated = annotated[..., ::-1]
        ax.imshow(annotated)
        ax.axis("off")
    plt.suptitle("Annotated Skin Lesion Predictions", fontsize=16)
    plt.tight_layout(rect=[0, 0.03, 1, 0.95])
    plt.show()
```